



Thank you for using Simio. We expect that you will be pleased with its power, flexibility, and ease of use.

We are pleased to bring you the latest update to **Simio Release 11**. Simio's patented processes provide ease of use while also allowing unprecedented flexibility without requiring any coding. And Simio leads the industry in data representation and analysis. Simio is unique in our ability to do design, planning, scheduling, and schedule risk-analysis in one application.

Many of our customers have cited Simio's flexibility, ease-of-use, and great support among the reasons why they have adopted Simio -- often switching so they can do things that are difficult or impossible with other products. But we are not resting on that success -- we are continuing to make Simio easier to use and more flexible, while adding leading the industry with more technology breakthroughs.

This document describes the wealth of new features that have been recently added. Simio is ready to take on your most demanding modeling tasks.

The desire to teach using the latest innovations, combined with Simio's comprehensive academic package are moving Simio towards becoming the most widely taught simulation package with over 800 leading academic institutions worldwide already adopting Simio.

We are anxious to hear about your experiences. Please post your feedback in the forums mentioned below or send it directly to me. Thanks again for choosing Simio.

Dave Sturrock
Vice President – Operations, dsturrock@simio.com

Simio Release Notes



Support

The best way to get support is to use the **Simio User's Forum** found at www.simio.com/forums, or contact us at support@simio.com using one of the links on the Support Ribbon. Sign up on the user's forum as a **Simio Insider** to get full access to find product information, a place to post problems and questions, and the opportunity to engage in discussions with other users and the Simio team.

You can also find Simio User's Groups on:

LinkedIn (<https://www.linkedin.com/groups/55167>) and

Facebook (www.facebook.com/groups/13863832711).

Please visit our simulation community resources web page (www.simio.com/resources.html) for additional information.

The **Support Ribbon** (the Support tab) of the Simio software contains shortcuts to the above, plus much more. Don't miss the links to our on-line documentation, training, videos, downloads, and other helpful resources. While Simio includes comprehensive on-line help available at the touch of "F1" or  in the product, the numerous books and free video training courses provide a great way to get started using Simio.

Unfortunately, we cannot provide support to students. We encourage students to work through their instructors, or to reach out to the broader user community using the Simio Insiders User Forum.

What is a Sprint?

We are following an agile methodology in which we develop in three week cycles called sprints. At the end of each sprint we have tested, documented, distributable software including new functionality. The following pages describe the enhancements added during each sprint. It is not **necessary** that you download and install every Sprint unless you are interested in a particular feature. But we encourage you to do so when convenient so that you are always working with the best software, documentation and examples we can deliver. Most sprints are posted to the Simio Insiders download area. The public releases (discussed below) are updated approximately quarterly.

Where to Find New Releases

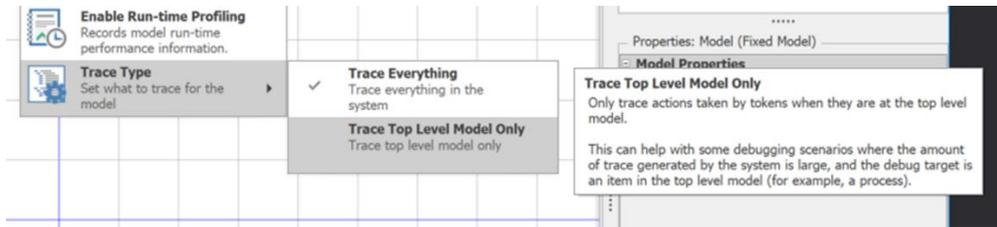
The latest public software releases can be found at: www.simio.com/download. Your activation will continue to work with new releases as long as your maintenance is current. Newer releases are often available on the [Simio Insiders forum](#).

Simio Release 11 – Sprint 190 – June 4, 2019

This sprint we have enhanced our trace options and added a shortcut for starting Simio64 and a Flow Library enhancement. Support for Sketchup 2019 and a Gantt search feature have also been added.

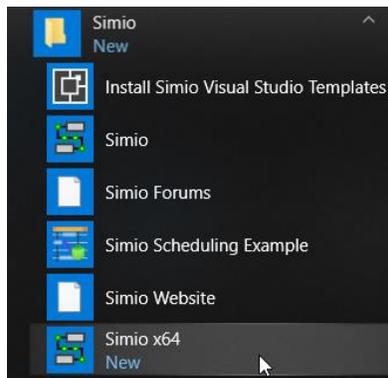
Trace Enhancement

There is a new “Trace Type” selection under the “Advanced Options” in the “Run Setup” group of the “Run” tab of the “Facility Tools” ribbon category. “Trace Everything” is the default. The selection is saved per model.



Install Option – 64 Bit Shortcut

We have added an install shortcut option to start Simio 64 bit (instead of 32 bit).



Flow Library Enhancement

We have added a new 'Auto Destroy Zero Volume Entities In Containers With Zero Rate Inflow' compatibility bit. This was done per a customer request. The default value of this setting is set to 'True', with models built prior to Sprint 190 having the setting of 'False'.

Sketchup 2019 Support

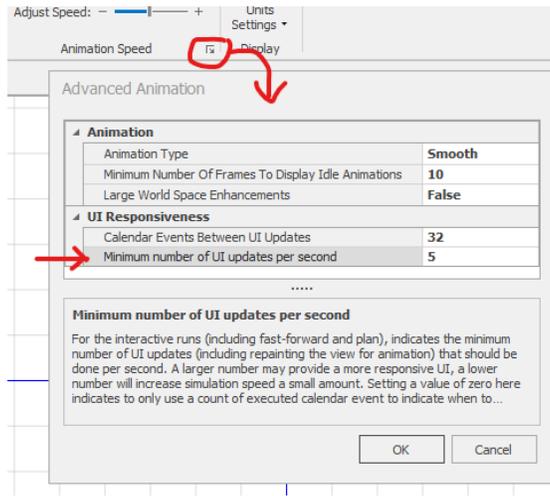
We have updated our Sketchup import to support Sketchup 2019.

Removed DirectX9 from Install

The DirectX9.0c folder is no longer in the Simio folder via C:\Program Files (x86)\Simio. It is still an option in the Application Settings *Graphics Type* (to support customers that still use DirectX9). We use DirectX11 by default.

UI Responsiveness

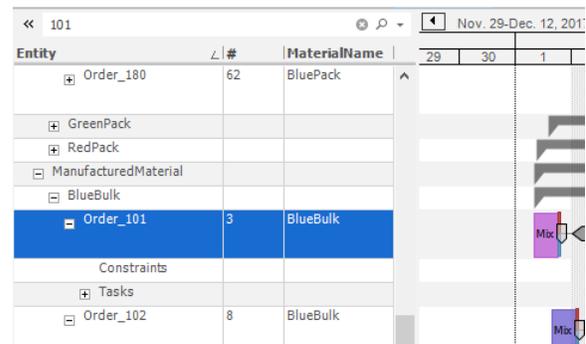
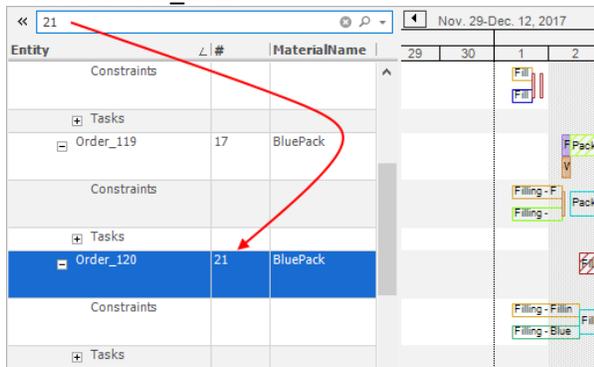
There is a new “Minimum number of UI updates per second” entry in the Advanced Animation dialog. It’s defaulted to 5 and provides much better UI responsiveness for otherwise unresponsive models.



Gantt Views – Search and Target Details

Gantt views now have a small search box at the top that does basic searching. If a user types or pastes a desired string in the search, Simio will go to the *first* row in the grid that contains that text. It also does partial matches.

In this example, searching for 21 matched a row containing that string, while searching for 101 finds the row for Order_101:



Also within the Planning tab, if a user double-click on an order/entity within the Target Detail tab, the user is taken to the same order within the Entity Workflow Gantt

Font Scale for Presenting/Teaching/Projection Scenarios

There is a new Font Scale setting in File -> Settings allowing you to scale up the font used across the application:

Display Deprecated Object Or El... **False**

Font Scale (percentage) **100**

Font Scale (percentage)

Indicates how much to scale the fonts used in various parts of the UI. A value of 100 is the default size. Using a larger size may help when using the UI on a projector in front of large groups.

So normally the UI looks like:

Properties: Server1 (Server)

Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selecti...	None
Transfer-In ...	0.0
Process Type	Specific Time
Processing Ti...	Random.Triangular(.1
Off Shift Rule	Suspend Processing

But if you set that scale to, say, 200 it would look like:

Properties: Server1 (Server)

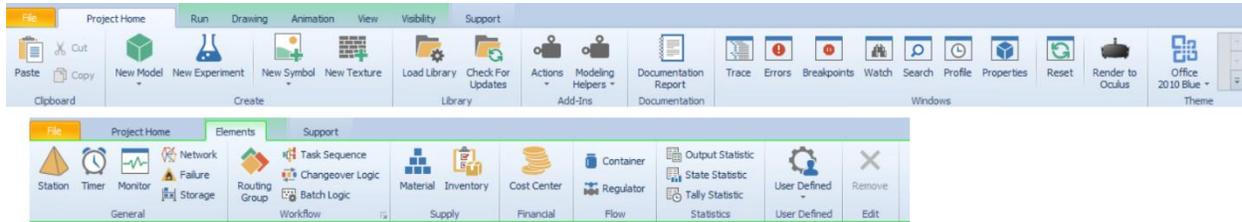
Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selectio...	None
Transfer-In Time	0.0
Process Type	Specific Time
Processing Time	Random.Triangular(.1
Off Shift Rule	Suspend Processing

Simio Release 11 – Sprint 189 – April 26, 2019

Welcome to our release version of Simio 11! This sprint, check out our new UI icons and skin, as well as a new SimBit displaying a few of the possible dashboards within experiments.

New Icons

We have updated our UI skin and icons on all the ribbons, tabs and panels, we hope you like the new look!



New SimBit

DashboardsWithinExperiments – This example is a modification to the SimBit ServerUsingTaskSequenceWithWorkers and includes an experiment where the number of workers and other properties can be changed. The new Dashboard Reports feature within Experiments has been used to include graphical dashboards that display response results either within a scenario or across scenarios.

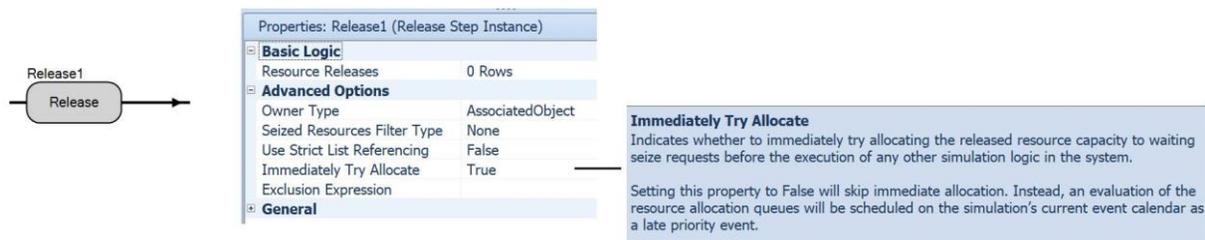
Simio Release 11 – Sprint 188 – April 18, 2019

Welcome to our pre-release version of Simio 11! This sprint, we have provided options for resource allocation queues when seizing/releasing the main object resources in Server / Combiner / Separator / Filler / Emptier, as well as the secondary resources.

Release Step Enhancement

A new *Immediately Try Allocate* Boolean property is now available for the Release step. Note that the default value of this property is True (the present behavior).

This feature is useful when multiple resources may be released simultaneously, as setting this property to False will allow entities in resource allocation queues to wait until all resources have been released before attempting to allocate a resource to an entity(s) based on any resource selection rules.



The diagram shows a 'Release1' step in a process flow, represented by a rounded rectangle with an arrow pointing right. To its right is a screenshot of the 'Properties: Release1 (Release Step Instance)' window. The window is divided into sections: 'Basic Logic' (Resource Releases: 0 Rows), 'Advanced Options' (Owner Type: AssociatedObject, Seized Resources Filter Type: None, Use Strict List Referencing: False, Immediately Try Allocate: True, Exclusion Expression), and 'General'. A callout box points to the 'Immediately Try Allocate' property, explaining its function.

Properties: Release1 (Release Step Instance)	
Basic Logic	
Resource Releases	0 Rows
Advanced Options	
Owner Type	AssociatedObject
Seized Resources Filter Type	None
Use Strict List Referencing	False
Immediately Try Allocate	True
Exclusion Expression	
General	

Immediately Try Allocate
Indicates whether to immediately try allocating the released resource capacity to waiting seize requests before the execution of any other simulation logic in the system.
Setting this property to False will skip immediate allocation. Instead, an evaluation of the resource allocation queues will be scheduled on the simulation's current event calendar as a late priority event.

Server / Combiner / Separator / Filler / Emptier Enhancement

For Server, Combiner, Separator, Filler, and Emptier objects, we have added a *Process Logic -> Other Processing Options* category in the Properties window with *Immediately Try Seize* and *Immediately Try Allocate When Released* properties.

By default, those properties are 'False' and 'True', respectively. Note that for any model saved in an earlier library object version, both those properties will be loaded with the value 'True' as that was the previous implicit behavior.

The default of *Immediately Try Allocate When Released* for the primary resource is set to 'True', so that by default the process logic will allow a new entity waiting for the Server (or other object type) to immediately seize it when the Server has been released by the previous entity, so as to potentially be first inserted into allocation queues for secondary resources before any secondary resource allocation attempts occur.

Properties: Server1 (Server)

Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Transfer-In Time	0.0
Process Type	Specific Time
Processing Time	Random.Triangular(1,2,3)
Off Shift Rule	Suspend Processing
Other Processing Options	
Immediately Try Seize	False
Immediately Try Allocate When Released	True

Immediately Try Seize
For an arriving entity, indicates whether to immediately try seizing the server before the execution of any other simulation logic in the system and, if successful, skipping the server's allocation queue.
Setting this property to False will just insert the seize request into the server's allocation queue. An evaluation of that queue will then be scheduled on the simulation's current event calendar as a late priority event.
NOTE: This property setting applies only if the server has an input buffer. Otherwise, if no input buffer, then an arriving entity will always immediately try to enter the server's processing station and seize it.

Immediately Try Allocate When Released
Once an entity has exited processing and released the server, indicates whether to immediately try allocating the released server capacity to waiting seize requests before the execution of any other simulation logic in the system.
Setting this property to False will skip immediate allocation. Instead, an evaluation of the server's allocation queue will be scheduled on the simulation's current event calendar as a late priority event.

For all the same objects, there are also the same new properties within the *Secondary Resources -> Resource for Processing* -> *Advanced Options* category as they relate to the secondary resource.

By default, those properties are both 'False'. Note that for any model saved in an earlier library object version, both those properties will be loaded with the value 'True' as that was the previous implicit behavior.

Secondary Resources

Resource for Processing	
Object Type	Specific
Object Name	
Selection Goal	Preferred Order
Request Move	None
Off Shift Rule	Suspend Processing
Advanced Options	
Selection Condition	
Immediately Try Seize	False
Keep Reserved If	
Reservation Timeout	
Immediately Try Allocate When Released	False

Immediately Try Seize
Indicates whether to immediately try seizing the resource before the execution of any other simulation logic in the system and, if successful, skipping the resource allocation queues.
Setting this property to False will just insert the seize request into the resource allocation queues. An evaluation of those queues will then be scheduled on the simulation's current event calendar as a late priority event.

Immediately Try Allocate When Released
Once an entity has finished processing and released the resource, indicates whether to immediately try allocating the released resource capacity to waiting seize requests before the execution of any other simulation logic in the system.
Setting this property to False will skip immediate allocation. Instead, an evaluation of the resource's allocation queue will be scheduled on the simulation's current event calendar as a late priority event.

For a Server, Combiner, or Separator object, if using the *Process Type* 'Task Sequence', within the *Resource Requirements -> Advanced Options* category in the Properties window of a processing task, there are new *Immediately Try Seize* and *Immediately Try Allocate When Released* properties.

By default, those properties are both 'False'. Note that for any model saved in an earlier library object version, both those properties will be loaded with the value 'True' as that was the previous implicit behavior.

Properties:

Task Information	
Sequence Number	10
Name	
Branch Type	Always
Process Type	Specific Time
Processing Time	Random.Triangular(1,2,3)
Auto Cancel Trigger	All Immediate Predecessors Cancelled
Resource Requirements	
Object Type	Specific
Object Name	
Selection Goal	Preferred Order
Request Move	None
Off Shift Rule	Suspend Processing
Advanced Options	
Number Of Objects	1
Units Per Object	1
Selection Condition	
Must Simultaneously Seize	False
Immediately Try Seize	False
Keep Reserved If	True
Reservation Timeout	Math.Epsilon
Immediately Try Allocate When Released	False

Immediately Try Seize

Indicates whether to immediately try seizing the task resource requirements before the execution of any other simulation logic in the system and, if successful, skipping the resource allocation queues.

Setting this property to False will just insert the seize request into the resource allocation queues. An evaluation of those queues will then be scheduled on the simulation's current event calendar as a late priority event.

NOTE: This property setting may vary from task to task but not from resource requirement to resource requirement for the same task. Thus, if using a table driven approach to define task sequence data, the value of this property may only be specified as a constant or by referencing a column in the table that defines each individual processing task.

Immediately Try Allocate When Released

Once the task has completed and released the resource requirements, indicates whether to immediately try allocating the released resource capacity to waiting seize requests before the execution of any other simulation logic in the system.

Setting this property to False will skip immediate allocation. Instead, an evaluation of the resource allocation queues will be scheduled on the simulation's current event calendar as a late priority event.

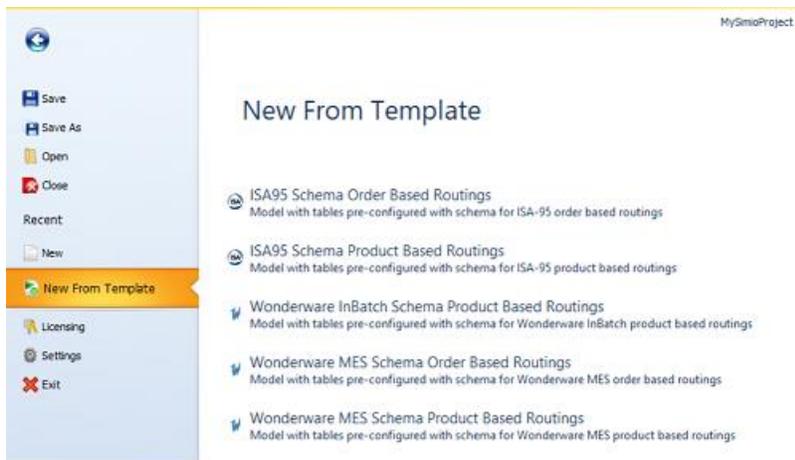
NOTE: This property setting may vary from task to task but not from resource requirement to resource requirement for the same task. Thus, if using a table driven approach to define task sequence data, the value of this property may only be specified as a constant or by referencing a column in the table that defines each individual processing task.

Simio Release 11 – Sprint 187 – April 4, 2019

Welcome to our pre-release version of Simio 11! This sprint, we have added Experiment Dashboard Reports for creating custom dashboards with experiment replications and/or scenarios, Project Recovery files in case of system outages, and Project Templates for opening models with existing data table structures and/or custom objects. We've also added runtime error information to our Trace window for easier debugging, and updated our Workstation based SimBits to instead utilize Server objects (Note: Workstation object is deprecated, see below).

New Project Templates

We have removed the Scheduling and Wonderware MES buttons found on the Content ribbon within the Data tab. Users can now use the File > New From Template option to view a selection of templates that are pre-installed with Simio:



Users may also create their own templates by just saving a project file, ideally with a Name, Icon, and Description set in the Project Properties, and saving it to the Project Templates location:
C:\Users\Public\Documents\Simio\ProjectTemplates.

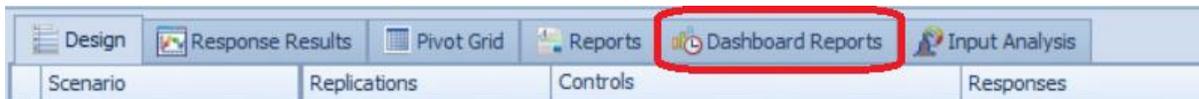
Trace Enhancement – Include RunTime Error Information

We now include runtime error information within the Trace window. For example, when an error within Simio occurs, users can now open the Trace window (even if Trace was not turned on) and see the Entity/Object/Process/Step where the error has occurred.

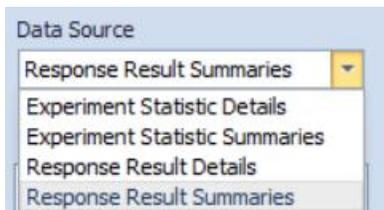
Input@Server1	OnEnteredToAssociatedObj...	1	[End]	Process 'Input@Server 1.OnEnteredToAssociatedObject' execution ended.
			[Begin]	Process 'Server 1.OnEnteredProcessing' execution started.
			[EndTransfer] Int...	Entity 'DefaultEntity.9' ending transfer into station 'Server 1.Processing'.
Server1	OnEnteredProcessing	2	[Decide] IfProcess...	Token branching on condition 'ProcessType==Enum.TaskProcessType.SpecificTime'. Token sent to 'True' exit.
			[Delay] Processing ...	Unable to get value of property 'Server 1.ProcessingTime'. Division by zero error occurred while evaluating expression.

Experiment Dashboard Reports

We have added the ability to generate Dashboard Reports within the Experiments of a project. This can be done by using the new Dashboard Reports tab:



This tab opens a view like the Dashboard Reports view for interactive runs. Under Experiments, however, there are a different set of data sources, as shown below, where the “Summaries” data sources include average/half-with/min/max across replications, and the “Details” data sources have the individual replication observations.

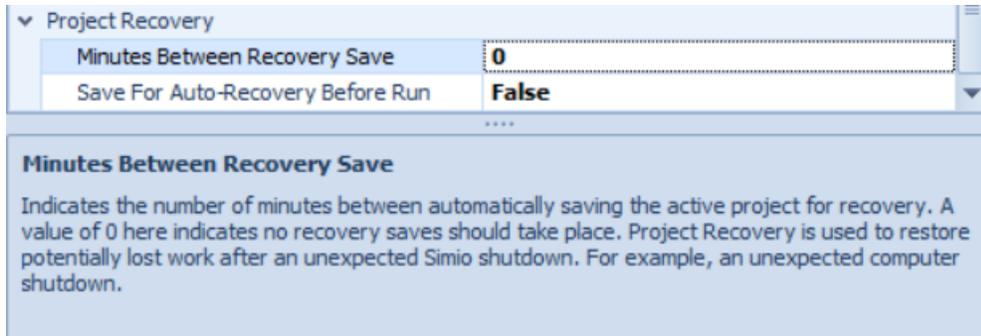


In the below graph, the HospitalEmergencyDepartment example was used, with additional Responses to capture ScheduledUtilization statistics for the Nurse and Doctor workers. Once the experiment scenarios were run, the Response Results Summaries were used to generate a ‘Chart’ type dashboard. Within the Chart’s Data Items, the ‘ResponseName’ is used within the *Series*, the ‘ScenarioName’ is specified under *Arguments* and the ‘Mean’ under the *Values* section. A Filter Element (List Box) was used for the filtering section on the right side of the dashboard below (‘ResponseName’ under *Dimensions* Data Items).

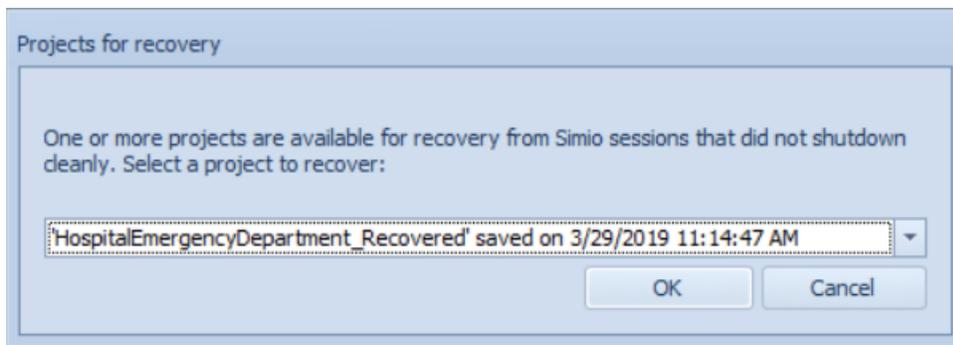


Project Recovery Files Option

In File > Settings within the Application Settings window, there are now two new settings for “Project Recovery”:



By default, this feature is off, as the *Minutes Between Recovery Save* is set to '0' and the *Save for Auto-Recovery Before Run* is set to 'False'. If a user was to turn one of those on to get a "Recovery Save" (wait the time limit or run), then end Simio using 'End Task' through the Task Manager (or other unexpected 'crash'), the next time a Simio instance starts up, the user will see:



Clicking OK opens the recovery save, saves it in a new file named [ProjectName]_Recovered, and deletes the recovery save. Clicking Cancel keeps the recovery save. Users can explicitly delete them by deleting the contents of the %localappdata%\Simio\Recovery directory, where %localappdata% is configured on most systems to be C:\Users\[user name]\AppData\Local.

Seize Step – Enhancement

The Seize step now includes an *Immediately Try Seize* property that allows the user to specify whether to immediately seize any available resource(s) or to allow other same time events within Simio's event calendar to occur prior to seizing any available resource(s). This feature is used within the 'Switch Resources If Possible' logic for secondary resources *Off Shift Rule* option.



MultiEchelonSupplyChain Example – Updated Description

The document and help descriptions of the *MultiEchelonSupplyChain* example have been updated with screen shots and additional descriptions of internal logic.

Workstation SimBits - Updated to Server SimBits

ServerWithMaterialConsumption - This new SimBit replaces the previous *WorkstationWithMaterialConsumption* SimBit. This new SimBit uses the Material Requirements section of properties within a Server Task Sequences to consume and produce materials required for processing.

ServerWithMaterialConsumptionAndReplenishment - This SimBit is a second model within the above *ServerWithMaterialConsumption* project and replaces the previous *WorkstationWithMaterialConsumptionandReplenish* SimBit. In addition to using the 'Task Sequence' type processing for material requirements, it also uses Monitor elements to track the material inventory and Produce steps within processes to handle material replenishment.

ScheduledMaterialArrivals – This updated SimBit includes now material consumption within the 'Task Sequence' type processing of a Server instead of within a Workstation object. A data file is used to specify material arrival times into the system.

ScheduledMaterialArrivalsWithDeviation – This updated SimBit now includes material consumption within the Task Sequences of a Server instead of within a Workstation object. The model shows how variation around the material arrival times affects a system.

UsingRelationalTables – This updated SimBit now uses Server objects in place of Workstation objects in the Facility window. The model includes data that exists in different tables and the tables are linked together with Foreign Keys, as in a relational database.

ServerWithSequenceDependentSetup – This new SimBit replaces the previous *WorkstationWithSequenceDependentSetup* SimBit. The Server uses 'Task Sequence' type processing to specify Sequence Dependent Setup between various part types in the system.

Financials – This updated SimBit now includes Server objects in place of Workstation objects. The model demonstrates how to calculate costs, such as capital costs of objects, usage costs, holding costs and the cost of transporting an entity on a vehicle.

RelationalTablesInRepeatingProperty – This updated SimBit now uses Servers in place of Workstations for processing multiple patient entity types through a system. Relational data tables are used to specify different processing times, as well as multiple doctors and/or nurses for processing, for the patient types.

Workstation Object within Standard Object Library – Deprecated

The Workstation object has now been deprecated from the Standard Object Library. Note that any models that include Workstation object instances will continue to see the Workstation object in the library. Existing models that do not contain any Workstations, as well as new Models, will not have that object visible within the library. See the section below for more details on how to enable a deprecated object if necessary.

Object Visibility Within a Library - Updated

The "Visible in Library" model level property has changed to "Library Visibility". A user can explicitly show/hide deprecated objects from the library right click menu (this menu item is only enabled for libraries with deprecated objects in them. Note that a deprecated object ***will*** be shown if it has any instances in the associated model, ***no matter*** the "Show Deprecated Objects" setting.

Advanced Options

Object Type	Fixed
Parent Class	Fixed
Keywords	
Categories	
Resource Object	False
Runnable	True
Library Visibility	Always
Table Imports are Undoable	Always
Load Action	Never
Update Interval	Deprecated
Check Base for Advanced...	False

Library Visibility

Indicates if this object is shown when the project is attached as a library.

"Always" indicates the object will always be shown.
 "Never" indicates the object will never be shown.
 "Deprecated" indicates that object will not be shown unless the active model uses this object, or the user has explicitly enabled "Show Deprecated Objects" in the library view.

Facility Processes Definitions Data

Libraries

- Standard Library
 - Source
 - Sink
 - Server
 - Combiner
 - Separator
 - Resource

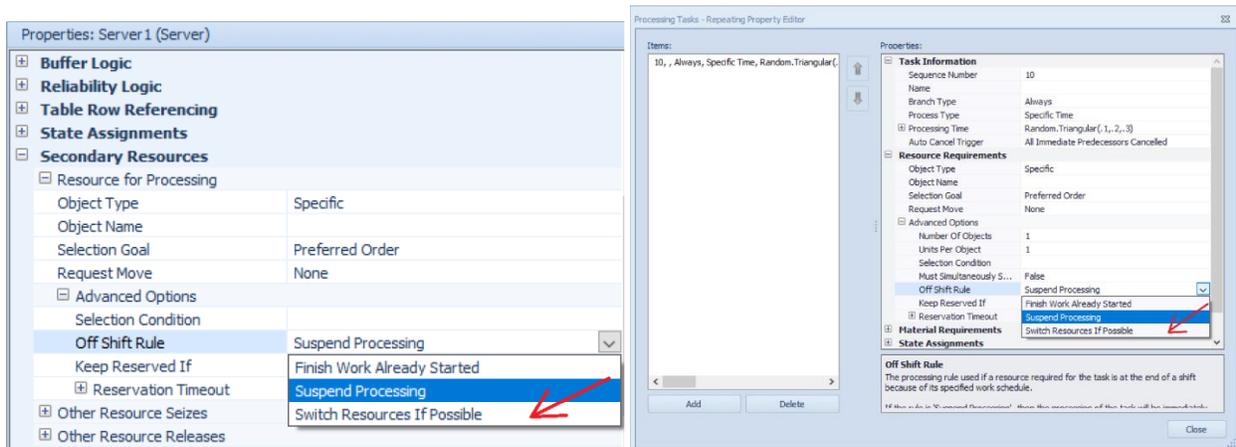
Subclass
 Load Library
 Unload Library
 Show Filter
 Show Deprecated Objects

Simio Release 10 – Sprint 186 – March 14, 2019

This sprint, we have enhanced the use of secondary resources with regards to shift changes, allowing the switching of resources between shifts. While this could always be done, the features below make it much easier! This has been a user-requested feature from multiple customers. We've also added an optional experimentation scenario timestamp, as well as several Simio RPS features.

Off Shift Rule For Secondary Resources – Switch Resources If Possible

Previously, when a secondary resource was used within an object for general processing or within task processing, the user had two options for *Off Shift Rule*, including 'Suspend Processing' and 'Finish Work Already Started'. We have now enhanced the *Off Shift Rule* to also include 'Switch Resources If Possible'. This will allow processing to continue when a shift change occurs, which changing the secondary resource. For example, FirstShiftWorker and SecondShiftWorker are members of a worker list. At a Server, if a process begins at 3:45 pm (during first shift) and lasts for 3 hours (into second shift), when the FirstShiftWorker goes offshift, Simio will seize a SecondShiftWorker (if available) to continue the processing the entity at the Server.



We have also made several supporting enhancements, as noted below, that allowed configuration of this feature.

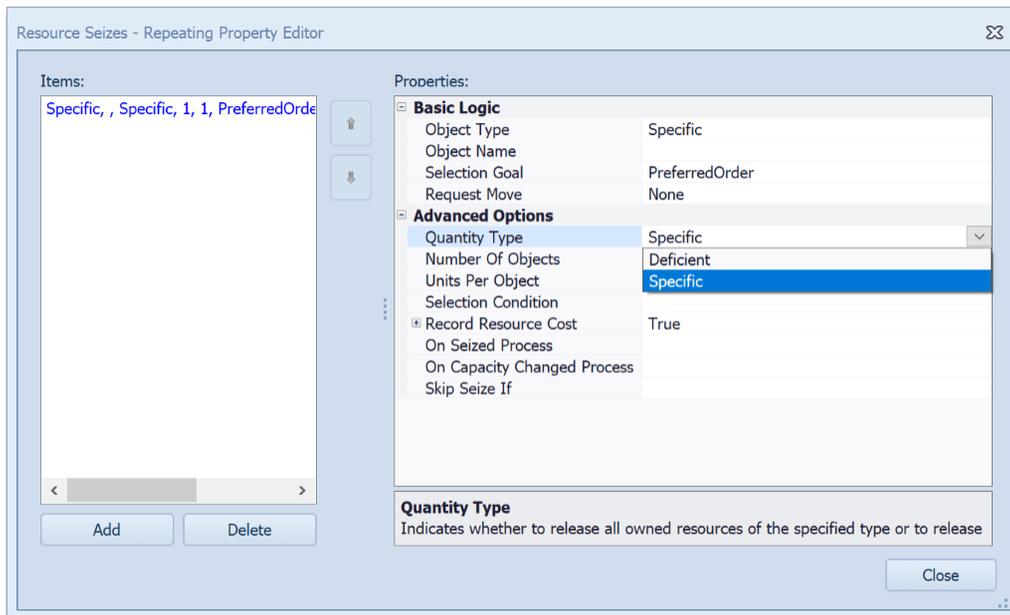
Task – New Function

We have added a new task function:

Task.PrimaryToken - Returns a reference to the token that is executing the task's primary process.

Seize and Release Step – Multiple Enhancements

To support the above 'Switch Resources If Possible' option for Off Shift Rule, we have added several properties to Seize and Release steps. First, within the Seize step, we have added a *Quantity Type* property (under Advanced Options) that indicates the type of resource quantity to seize. The options are 'Specific', which represents the exact quantities entered in the *Number Of Objects* and *Units Per Object* properties and 'Deficient', which is derived by subtracting the already seized resources of the specified resource type from the quantities entered in the *Number Of Objects* and *Units Per Object* properties. For example, if the *Number Of Objects* property is specified as '5' and the number of already seized resources that meet the requirement is '3', then the deficient number of resource objects to seize is calculated to be '2'.



Then, within both the Seize and Release steps, we have added a *Seize Resources Filter Type* property that can be set to 'None' or 'Token'. With the 'Token' option, an additional property named *Filter Token Reference* can be specified. To see how these new properties are used, please review the 'OnSecondaryResourceCapacityChanged' process within a sub-classed Server or review the Simio Help for these steps.

Properties: Seize1 (Seize Step Instance)	
Basic Logic	
Resource Seizes	0 Rows
Must Simultaneously Seize	False
Advanced Options	
Owner Type	AssociatedObject
Seized Resources Filter Type	None
Exclusion Expression	
General	

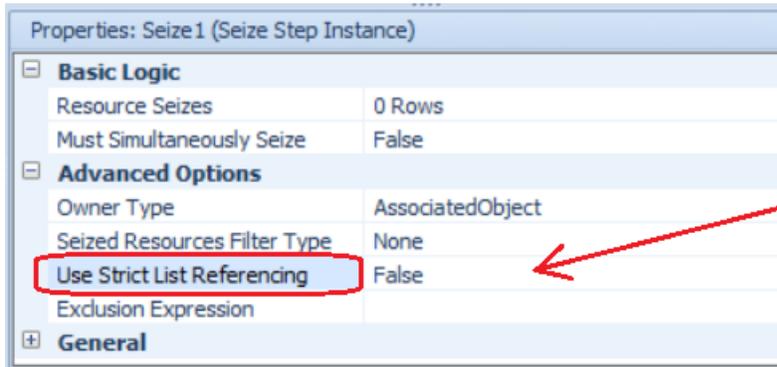
Properties: Release1 (Release Step Instance)	
Basic Logic	
Resource Releases	0 Rows
Advanced Options	
Owner Type	AssociatedObject
Seized Resources Filter Type	None
Exclusion Expression	
General	

Properties: Seize1 (Seize Step Instance)	
Basic Logic	
Resource Seizes	0 Rows
Must Simultaneously Seize	False
Advanced Options	
Owner Type	AssociatedObject
Seized Resources Filter Type	Token
Filter Token Reference	
Exclusion Expression	
General	

Properties: Release1 (Release Step Instance)	
Basic Logic	
Resource Releases	0 Rows
Advanced Options	
Owner Type	AssociatedObject
Seized Resources Filter Type	Token
Filter Token Reference	
Exclusion Expression	
General	

Unrelated to the 'Switch Resources If Possible' enhancement, the Seize and Release steps now include a *Use Strict List Referencing* property. For the Seize step, if seizing from list and quantity type is 'Deficient', this indicates whether the already seized resources must have been seized from the same specified list, instead of merely being members of the list. For the Release step, if releasing from list, this indicates

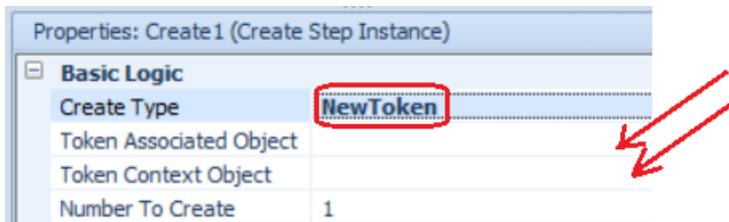
whether the released resource(s) must have been seized from the same specified list, instead of merely being members of the list. This is particularly useful when resources are members of multiple lists.



Create Step – Enhancement

Within the Create step, we have added the option to create a new token into the process. The Create step may now be used to create new entity objects of a specified type, create copies of an existing entity, or to simply create new tokens that reference existing objects. New tokens associated with the created or referenced objects will exit the 'Created' exit point of the step.

When a new token is created using the Create step, the user can specify the Token Associated Object and/or the Token Context Object. If not specified, the associated object reference and/or context object reference of the original token executing the Create step will be assumed.



Allocating Resources When Capacity Increased - Advanced Compatibility Setting Change

In Run Setup -> Advanced Compatibility Settings, the default value for new models of *the Schedule Late Current Event To Try Allocating When Resource Capacity Increased* compatibility setting has been changed from 'False' to 'True'.

This setting indicates whether to schedule a late priority current event to try allocating whenever the capacity of a resource has been increased. Otherwise, allocation will be immediately attempted before the execution of any other simulation logic in the system.

In the Standard Library, the Vehicle and Worker objects OnCapacityChanged process logic has also been modified to delay for epsilon time when coming on-shift before any attempt to allocate the vehicle or worker resource.

If using the 'Switch Resources If Possible' *Off Shift Rule* for secondary resource requirements at a Server, this change now means that the default behavior of that feature is WIP tasks attempting resource switching will have priority seizing secondary resources coming on-shift before any new tasks that have been waiting in the resource allocation queues are able to try to seize.

Seize Selection Goal - Enhancement

In Run Setup -> Advanced Compatibility Settings, an *Always Use Seize Selection Goal When Checking Resource Allocation Queue* compatibility setting has been added. This setting indicates whether a Seize step's selection goal should be used not only at the initial seize attempt but also whenever any resource allocation queue checks occur - if waiting for resources is necessary - regardless of the *Must Simultaneous Seize* option setting. The recommended value for this setting is 'True' but will be set at 'False' for older models built prior to Sprint 186.

In a simple example, suppose an entity is waiting to Seize from a worker list, but all the workers are currently off-shift. Thus, the entity waits in all the allocation queues of the resource candidates. Then suppose all the workers in the list come on-shift.

In the old software behavior - if the above compatibility setting is left at 'False' for an old model - whichever candidate resource's shift change happened to be processed first - usually based on the order that objects were placed in the model - was always the resource seized. The *Selection Goal* (i.e., resource selection rule) specified on the Seize step would be ignored. Now, what happens is regardless of which candidate resource happens to be checking its allocation queue, an entity always try to seize the best candidate resource per the specified *Selection Goal* on the Seize step. In the simple worker example described above, all the workers come on-shift and then the entity seizes the 'best' one per whatever resource *Selection Rule* has been specified. This has always been the way things worked if using the *Must Simultaneously Seize* option, but now that behavior is consistent across the board.

Experimentation - Scenario Timestamp

We have added the ability to add a Scenario timestamp column to the experiment scenarios. To view this column, right Click on a column header (such as Status, Required, or any column header) in the Experiment View. Select 'Column/Band Chooser' and a small box appears. In the box, double click 'Last Run Completed At'.

The screenshot shows the 'Experiment View' table with the following data:

Scenario	Name	Status	Replications	Staff - Controls	Responses	Average Length Of Stay	Average Leave Without Being Seen
001	Complete	Sort Ascending	4	4	40080.8	42.5378	16.1163
002	Complete	Sort Descending	3	3	31286.4	34.3743	96.6289
003	Complete	Hide This Column	3	4	38515	31.1299	118.964
004	Complete	Column/Band Chooser	4	3	32892.8	45.2231	23.2924

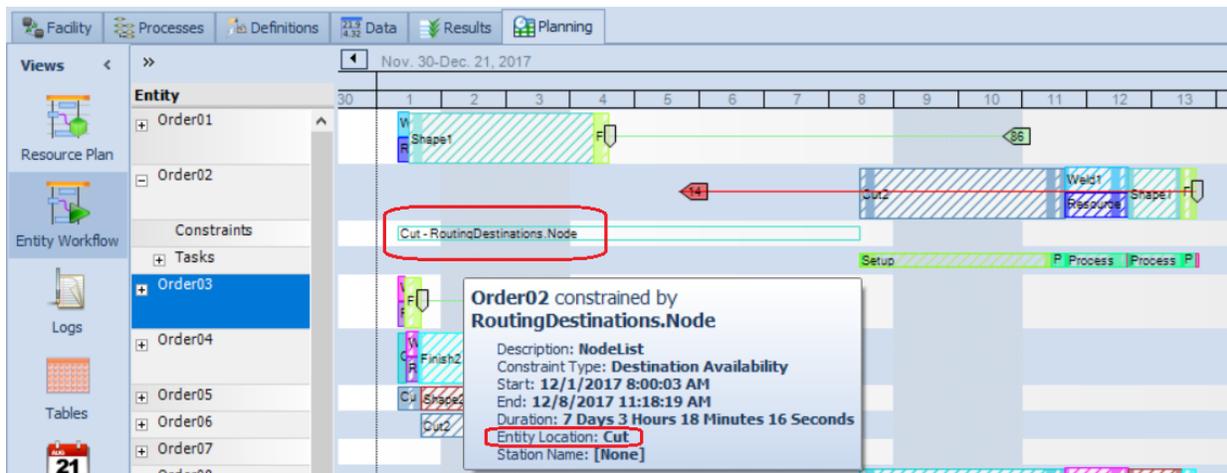
The 'Customization' dialog box shows the 'Last Run Completed At' option selected.

It now appears next to Status column. To hide the column, Right Click 'Last Run Completed At' and select 'Hide This Column'.

Scenario	Replications	Staff - Controls	Responses			
Name	Status	Last Run Completed At:	Of Nurses	Number Of Doctors	Cost Of Staff	Average Length Of Stay
001	Completed	3/9/2019 10:08:20 AM	4	4	40080.8	42.5378
002	Completed	3/9/2019 10:08:18 AM	3	3	31286.4	34.3743
003	Completed	3/9/2019 10:08:20 AM	3	4	38515	31.1299
004	Completed	3/9/2019 10:08:19 AM	4	3	32892.8	45.2231

Entity Gantt – Constraint Destination Enhancement (RPS Edition)

We have updated the destination availability constraints information on the Entity Gantt now shows the name of the parent item where entities are queued. The Constraint Log column named Facility Location now tracks the entity location.



New Resource Info Log (RPS Edition)

There is a new Resource Info Log that holds one record per resource used in a model. It has Resource Id and Resource Name information, Display Category, and optionally one or more user expression columns. The expression columns can be matched to Portal traits, like expressions in the Resource Usage Log. It's important to note that filtering the Resource Info Log by trait does not filter other logs.

Simio Release 10 – Sprint 185 – February 18, 2019

This sprint, we have enhanced the Table-Based Work Schedules to include exception tables for defining global calendars. OptQuest for Simio now evaluates both enumerator and list type properties in experimentation scenarios. Additionally, a new IModelHelperAddIn interface has been added to SimioAPI.Extensions.

Table Based Work Schedules – Exceptions

Within the Table Based Work Schedules, we have added the ability to specify an Exceptions table and associated data. Data within the Exceptions table will take precedence over the data within the Table work schedule. This is useful for defining multiple schedules that reference a 'global' exceptions table that specifies holidays, weekends, etc.

For example, in the below diagram, the data within the Exception table named TableExceptions (in red) will override the Table2 data for 6/29/2010 (in black with X).

The screenshot illustrates the configuration of a Table-Based Work Schedule. The Properties window for 'Schedule2' shows the following settings:

General	
Name	Schedule2
Table Name	Table2
Start Date Time	Table2.StartTime
End Date Time	Table2.EndTime
Value	Table2.Value
Cost Multiplier	1
Exception Table Name	TableExceptions
Exception Start Date Time	TableExceptions.StartTime
Exception End Date Time	TableExceptions.EndTime
Exception Value	TableExceptions.Value
Exception Cost Multiplier	0

Below the Properties window, two tables are shown:

Table2	Table Exceptions		
Start Time	End Time	Value	
1	6/28/2010 12:00:00 AM	6/28/2010 12:30:00 AM	1
2	6/28/2010 1:00:00 AM	6/28/2010 2:00:00 AM	1
3	6/28/2010 2:30:00 AM	6/28/2010 4:00:00 AM	1
4	6/28/2010 4:30:00 AM	6/29/2010 12:00:00 AM	1
5	6/29/2010 12:00:00 AM	6/29/2010 12:30:00 AM	1
6	6/29/2010 1:00:00 AM	6/29/2010 2:00:00 AM	1
7	6/29/2010 2:30:00 AM	6/29/2010 4:00:00 AM	1
8	6/29/2010 4:30:00 AM	6/30/2010 12:00:00 AM	1

Table2	Table Exceptions	Start Time	End Time	Value	Description
1		6/29/2010 12:00:00 AM	6/30/2010 12:00:00 AM	0	Holiday

The Value (Capacity) of an object using this Schedule2 will be 0 (Off for Holiday) on 6/29/2010 for 24 hours.

It's important to note that any Work Day Exceptions or Work Period Exceptions data specified within the object instance will take precedence over the Exceptions table data.

Experimentation - OptQuest Update

Within the experiment of model, if the user has the OptQuest add-in, it now recognizes has any Enum properties or List properties defined as controls. In the example below, the *ListProperty1* is a string list, consisting of East, West, North and South. The *EnumerationProperty1* is a selection type enum.

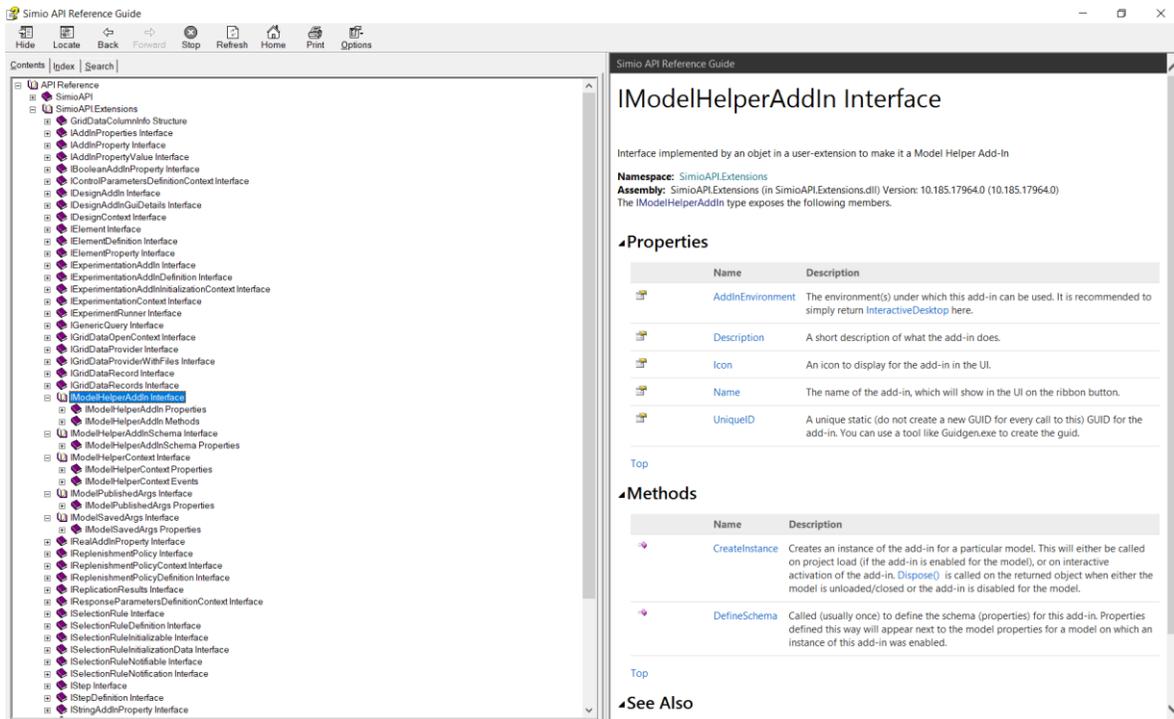
Design Response Results Pivot Grid Reports Input Analysis						
Scenario			Replications		Controls	
<input checked="" type="checkbox"/>	Name	Status	Required	Completed	EnumerationProperty1	ListProperty1
<input checked="" type="checkbox"/>	1	Idle	10	0 of 10	SmallestDistance	East
<input checked="" type="checkbox"/>	*					East North South West

OptQuest will then simply choose different combinations of these values as part of the optimization.

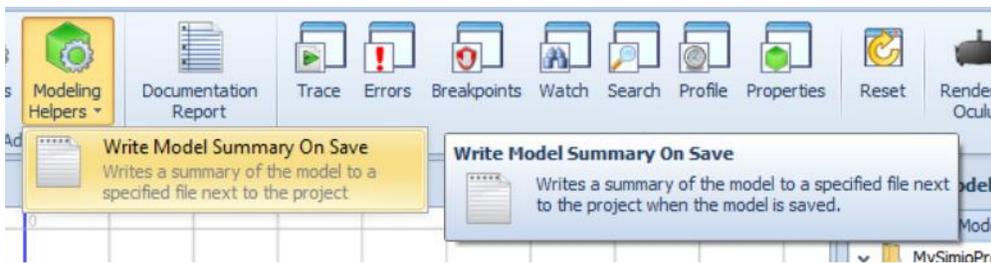
Design Response Results Pivot Grid Reports Input Analysis						
Scenario			Replications		Controls	
<input checked="" type="checkbox"/>	Name	Status	Required	Completed	EnumerationProperty1	ListProperty1
<input checked="" type="checkbox"/>	001	Running	6	1 of 6	Cyclic	South
<input checked="" type="checkbox"/>	002	Running	6	1 of 6	SmallestDistance	East
<input checked="" type="checkbox"/>	003	Running	6	1 of 6	LargestValue	West
<input checked="" type="checkbox"/>	004	Running	6	1 of 6	PreferredOrder	North
<input checked="" type="checkbox"/>	005	Running	6	1 of 6	SmallestValue	South
<input checked="" type="checkbox"/>	006	Running	6	1 of 6	SmallestValue	East
<input checked="" type="checkbox"/>	007	Running	6	1 of 6	SmallestDistance	West
<input checked="" type="checkbox"/>	008	Pending	6	1 of 6	SmallestDistance	South
<input checked="" type="checkbox"/>	009	Pending	6	1 of 6	PreferredOrder	West
<input checked="" type="checkbox"/>	010	Running	6	0 of 6	Cyclic	East
<input checked="" type="checkbox"/>	011	Running	6	0 of 6	LargestValue	South
<input checked="" type="checkbox"/>	012	Running	6	0 of 6	LargestDistance	North

IModelHelperAddIn Interface

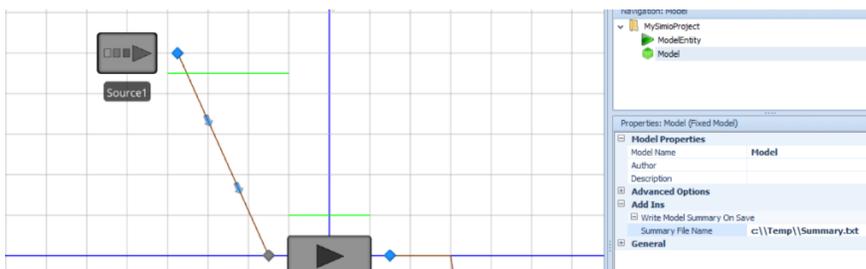
A new IModelHelperAddIn interface has been added to SimioAPI.Extensions. With this interface, there are methods to add code when the mode is loaded (CreateInstance) and when the model is closed (Dispose). There are also events handlers in the interface that can be used to add code when the model is saved and when the schedule is published to / or within Simio Portal.



When you install Simio, there is a code example of IModelHelper that gets installed at:
C:\Users\Public\Documents\Simio\Examples\UserExtensions\WriteModelSummaryOnSave
 If a user has an RPS license, there is a Model Helpers menu on the Project Home ribbon.



Once enabled, there is a property added to the model. You might need to click on Model Entity and then back to Model to get the properties to show.



The file from this example is written after the model is saved.

Simio Release 10 – Sprint 184 – January 25, 2019

This sprint, we have completed the Combiner and BatchLogic enhancements with options for releasing batches early. This also includes an updated SimBit illustrating these release early concepts. We've also added input argument support for the various Inventory element processes, as well as simplified the free space logic steps within ModelEntity and Worker/Vehicle objects.

BatchLogic Element and Standard Library Combiner – Release Batch Early Triggers

For a BatchLogic element or Standard Library Combiner, you can now specify options for releasing batches early, either based on time or events occurring in the system.

Release Batch Early Triggers Repeat Group Properties

The image shows a screenshot of the Simio software interface. On the left, a 3D model of a 'Combiner1' is shown. To its right is a 'Properties: Combiner1 (Combiner)' window. The 'Batching Logic' section is expanded, showing various options. A callout box points to the 'Release Batch Early Triggers' property, which is currently set to '0 Rows'. Below this, two 'Release Batch Early Triggers - Repeating Property Editor' windows are shown. The first window shows a 'Time Based' trigger configuration with 'Wait Duration' set to 0.0 and 'Release Decision Type' set to 'Always'. The second window shows an 'Event Based' trigger configuration with 'Triggering Event Name' set to 'Conditional' and 'Release Condition Or Probability' set to 0.0. Callout boxes provide definitions for 'Trigger Type', 'Wait Duration', 'Release Decision Type', 'Triggering Event Name', and 'Release Condition Or Probability'.

Release Batch Early Triggers
Optional time or event-driven triggers that can cause a parent entity to be released early before collecting its full target number(s) of member entities.

Trigger Type
The type of trigger.
A time based trigger can release a batch early once a specified wait duration expires.
An event based trigger can release a batch early whenever a specified event occurs.

Wait Duration
The wait duration until deciding whether to release a batch early.

Release Decision Type
The method used when the trigger has fired to decide whether to release a batch early.

Triggering Event Name
The name of the event whose occurrence will trigger a decision whether to release a batch early.

Release Condition Or Probability
The release condition or probability specified as an expression. If a probability then enter the chance of releasing the batch early as a value between 0.0 (0%) and 1.0 (100%).

The *Release Batch Early Triggers* defined for a parent entity can be either time based (trigger fires when specified wait duration expires) or event based (trigger fires when a specified event occurs). By default, the release decision when a trigger has fired is 'Always' but may alternatively be 'Conditional' or 'Probabilistic'.

SimBit - Updated

CombinerReleasingBatchEarly - This new SimBit replaces the previous RenegingCombiner SimBit. This new SimBit simply uses the *Release Batch Early Triggers* feature instead of a parent input buffer renege

trigger approach with a standalone extra node that then reroutes the entity back into the combiner, extra state assignments to recalculate available batch quantities, etc.

Inventory Element – Input Argument Support for Processes

Within the Inventory element, there are three different processes that may be specified, including *On Replenishment Order Process*, *On Balked At Backorder Process* and *On Reneged Backorder Process*. We have added support for Input Arguments from those processes.

Updated Free Space Movement Logic – ModelEntity/Worker/Vehicle

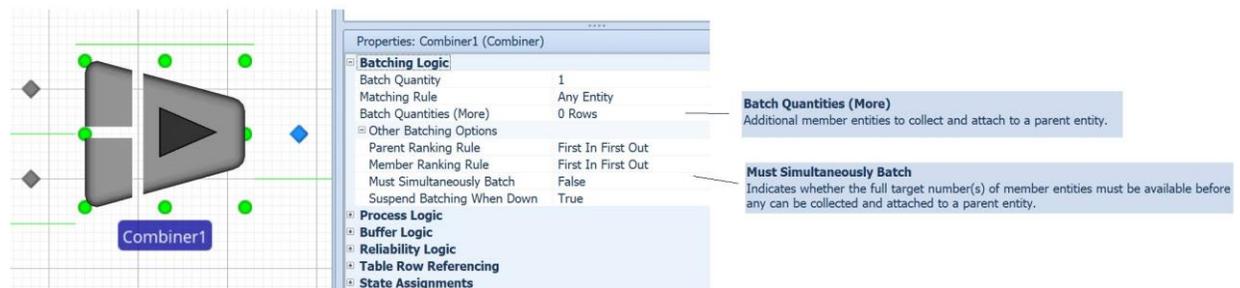
Within the *OnEnteredFreeSpace* processes of the *ModelEntity*, *Worker* and *Vehicle*, we have condensed the number of steps used from 15 to 7. This was done primarily for the Personal Edition limits, as each step within *ModelEntity* objects counts towards the limited number of steps allowed.

Simio Release 10 – Sprint 182/183 – January 3, 2019

This sprint, we have added several user-requested features, including multiple entity type batching within the Combiner and BatchLogic element, updated SimBit for batching multiple entity types, and new Input Parameters sampling options.

BatchLogic Element and Standard Library Combiner – New Properties

For a BatchLogic element or Standard Library Combiner, you can now specify a repeat group of batch quantity requirements *Batch Quantities (More)*, that is required to be collected and attached to each parent entity. Additionally, a *Must Simultaneously Batch* property has been added.



For example, suppose at a Combiner, the parent entities represent pallets and the following quantities and entity types are required to be packed onto each pallet:

Entity Type	Batch Quantity
A	1
B	3
C	2
D	1

In the Batching Logic properties of a Combiner, you can now easily model the above example by simply specifying the *Batch Quantity & Matching Rule* information as a repeating list of requirements, like this:

Batch Quantity	Matching Rule	Member Match Expression	Parent Match Expression
1	Match Members And Parent	ModelEntity.EntityType	A
3	Match Members And Parent	ModelEntity.EntityType	B
2	Match Members And Parent	ModelEntity.EntityType	C
1	Match Members And Parent	ModelEntity.EntityType	D

There is also a new Boolean property (*Must Simultaneously Batch*) that indicates whether the full target number(s) of member entities must be available before any can be collected and attached to a parent entity. If this property is 'False', then batch members will be collected from the batch logic's MemberQueue (and the Combiner's MemberInputBuffer) as soon as they become available. The parent entity eventually gets released once the target batch quantity or quantities are reached.

If this property is 'True', then no batch members can be collected by a parent entity until its full target batch quantity or quantities are available. In that case, the collection of all batch members and release of the parent entity is always going to happen as a single event (at a single point in time). Note that one possible result if forcing simultaneous batching is parent entities may jump over other parent entities in the parent queue if requirements differ between parents. For example, if there is only 1 member entity

waiting but the parent entity at the front of the queue needs 2 member entities, then that parent entity will have to wait until 2 are available before collecting any. However, if a parent entity behind it in the queue needs only 1 member entity, then that parent will be able to immediately collect it thus jumping the queue ranking.

BatchLogic and Combiner – Enhancements for Table Referencing

Previously, the *Matching Rule* property for the Batch Logic/Combiner was read in once at initialization as a single hard-coded setting. Thus, every parent entity had to use the same Matching Rule. Now, all of the *Batch Quantity* and *Matching Rule* information can be parent entity dependent (presumably coming from data table). For example, you can have one parent entity that needs to collect 10 Batch Quantity using *Matching Rule 'Any Entity'*, while another parent entity at the same Combiner needs to collect some batch quantities using *Matching Rule 'Match Members And Parent'* and so forth. Note that the *Must Simultaneously Batch* option can also be parent-entity dependent data.

BatchLogic Trace Enhancements

The BatchLogic related trace messages have been enhanced to be similar to the level of detail currently provided by the resource seizing, material/inventory usage, and routing group related trace.

An example of the new trace is below, showing the type of detail now provided.

```
-----  
Entity 'A.29' waiting to be batched by batch logic 'Combiner1.Batching'.  
Entity 'Pallet.28' waiting to collect '1' batch member(s) from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] ModelEntity.EntityType [AndParent] A [Matching Value = A]'.  
Entity 'Pallet.28' waiting to collect '3' batch member(s) from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] B [Matching Value = B]'.  
Entity 'Pallet.28' waiting to collect '2' batch member(s) from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] C [Matching Value = C]'.  
Entity 'Pallet.28' waiting to collect '1' batch member(s) from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] D [Matching Value = D]'.  
Batch logic 'Combiner1.Batching' checking parent queue which contains '1' item(s).  
Entity 'Pallet.28' found '1' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] ModelEntity.EntityType [AndParent] A [Matching Value = A]'.  
Entity 'Pallet.28' found '0' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] B [Matching Value = B]'.  
Entity 'Pallet.28' found '0' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] C [Matching Value = C]'.  
Entity 'Pallet.28' found '0' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] D [Matching Value = D]'.  
Batch logic 'Combiner1.Batching' selecting entity 'Pallet.28' waiting at rank '1' in parent queue.  
Entity 'Pallet.28' collecting batch member 'A.29' from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] ModelEntity.EntityType [AndParent] A [Matching Value = A]'.  
Entity 'B.30' waiting to be batched by batch logic 'Combiner1.Batching'.  
Batch logic 'Combiner1.Batching' checking parent queue which contains '1' item(s).  
Entity 'Pallet.28' found '1' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] B [Matching Value = B]'.  
Entity 'Pallet.28' found '0' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] C [Matching Value = C]'.  
Entity 'Pallet.28' found '0' currently available batch member(s) waiting in queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] D [Matching Value = D]'.  
Batch logic 'Combiner1.Batching' selecting entity 'Pallet.28' waiting at rank '1' in parent queue.  
Entity 'Pallet.28' collecting batch member 'B.30' from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] Entity.EntityType [AndParent] B [Matching Value = B]'.  
-----
```

SimBit - Updated

CombineMultipleEntityTypesOntoPallets - this new SimBit replaces the previous **CombineMultiplePartsTaskSequence**. This SimBit essentially models the same example, but now simply uses the new *Batch Quantities* repeat group on the Combiner to define the batch quantity requirements. Task sequences, extra process logic, data tables, etc. are no longer required to model this sort of problem.

Data Table – Row Editing in Property Grid

Within the Facility window, when a user selects an object or multiple objects, the property grid is provided for data input. We have now added this capability to **data table row editing**. When a single or multiple row are selected within a data table, the property grid now displays the column data for a single row or common data for multiple rows. This enables the user to multi-select (or filter to) rows and more easily edit multiple fields.

Order Id	Customer	Material Name	Release Date	Due Date	Order Status	Priority	Quantity	Adjusted Release Date
Order_SG_1	Walmart	EcoBikeSTDGreen	12/4/2017 8:00:00 AM	12/6/2017 4:00:00 PM	New	1	12	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_SW_1	Walmart	EcoBikeSTDWhite	12/4/2017 8:00:00 AM	12/6/2017 4:00:00 PM	New	1	10	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DG_1	Walmart	EcoBikeEXTGreen	12/4/2017 8:00:00 AM	12/6/2017 4:00:00 PM	New	1	8	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DW_1	Walmart	EcoBikeEXTWhite	12/4/2017 8:00:00 AM	12/6/2017 4:00:00 PM	New	1	9	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_SG_2	Ambridge Bike Shop	EcoBikeSTDGreen	12/4/2017 8:00:00 AM	12/7/2017 4:00:00 PM	New	3	15	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_SW_2	Ambridge Bike Shop	EcoBikeSTDWhite	12/4/2017 8:00:00 AM	12/7/2017 4:00:00 PM	New	3	9	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DG_2	Ambridge Bike Shop	EcoBikeEXTGreen	12/4/2017 8:00:00 AM	12/7/2017 7:00:00 PM	New	3	10	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DW_2	Ambridge Bike Shop	EcoBikeEXTWhite	12/4/2017 8:00:00 AM	12/7/2017 10:00:00 AM	New	3	11	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_SG_3	Costco	EcoBikeSTDGreen	12/4/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	2	9	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_SW_3	Costco	EcoBikeSTDWhite	12/4/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	2	12	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DG_3	Costco	EcoBikeEXTGreen	12/4/2017 8:00:00 AM	12/8/2017 12:00:00 PM	New	2	11	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_DW_3	Costco	EcoBikeEXTWhite	12/4/2017 8:00:00 AM	12/8/2017 4:30:00 PM	New	2	14	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP
Order_FRM_1	Stock	Frame	12/4/2017 8:00:00 AM	12/5/2017 4:00:00 PM	WIP	1	90	Math.If(ManufacturingOrders.OrderStatus == List.OrderStatus.WJP

Note that multi-selected rows will display data that is common between them.

Data Table – Data Binding Import Enhancements

Within data tables, when a table is bound to a file, a message will be shown displaying the time of the last import (X days, Y hours and Z minutes ago). If the data table is bound to a file, but data not yet imported, that will be noted as well.

Resource Name	Description	XLocation (Meters)	ZLocation (Meters)	Object Type	Work Schedule	CostRate (Default Currency per Hour)
Cut	Cut	-8		SchedTransferNode	StandardWeek	0
Cut1	Cut1	-4		SchedServer	StandardWeek	20

Resource Name	Node

Also, within data tables, there is now an option to import the bound data upon loading the simulation model. This option is available with Simio RPS only.

Binding Options

- Automatic**
Import table data at start of run.
- Manual**
Import table data manually.

Enable manual import for Operational Planning
Allow Operational Planning users to manually import this table's data.

Enable Append And Update Import Type
Enables or disables Append and Update import type.

Enable Table Import On Model Load
Enables or disables automatically importing table data on model load.

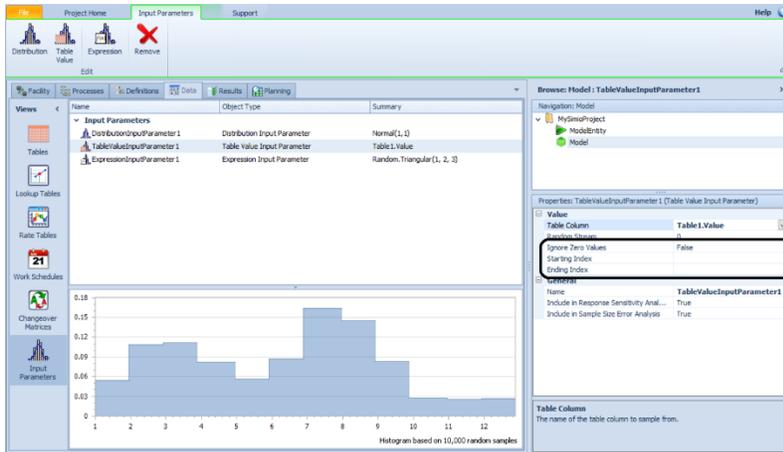
Enable Table Import On Model Load
If enabled, on model load, the table data will automatically be imported from the associated binding.

Enable Automatic Bound Table Importing
Enables or disables automatic importing of all bound tables at the start of a run.

Input Parameters – Table Value Enhancements

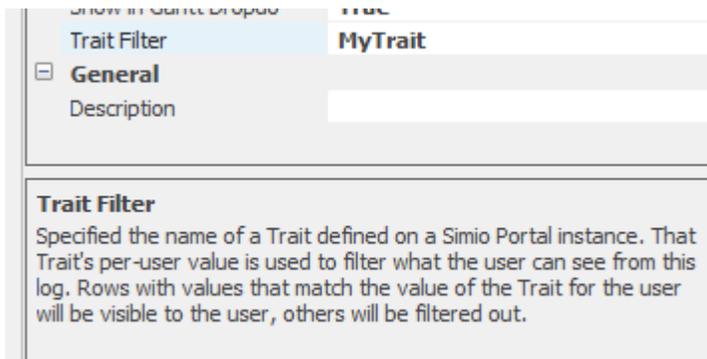
Within Input Parameters, when an input parameter is based on a Table Value, the user now has more flexibility on what data within the table data is sampled. There is a new *Ignore Zero Values* property that may be used to filter out '0' values from the table. If this property is set to 'True' and '0' is sampled, the simulation will simply resample.

Additionally, the user may specify a *Starting Index* (which defaults to 1) and/or *Ending Index* (which defaults to all). Simio will then only sample from that subset of rows.



Resource Usage and Task Logs - Enhancement

There is a new *Trait Filter* property for the expression columns for the Resource Usage and Task logs:



Within Simio Portal given the above example, a trait would be defined called 'MyTrait'. Each user can then have their own value (or comma separated values) assigned to 'MyTrait'. If the value of all expression columns with Trait Filters assigned in a row of the log matches the values for the user's trait(s), then they can see that row, otherwise they can't.

The row filtering works for viewing the Logs, using them as data sources for the Dashboards, and the Gantt data as well. The filtering is "filtering in", meaning not assigning a trait value for a user, or not even having a referenced trait of that same name defined in the portal, will filter away all rows for that log.

Project Home Ribbon – Modeling Helper Add-Ins – RPS Only

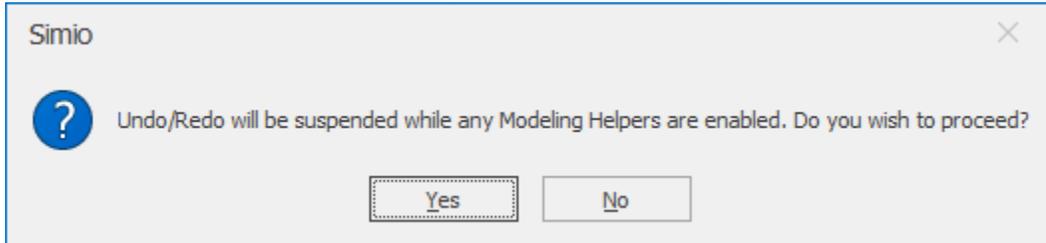
Users can now author design-time 'Modeling Helper' Add-Ins. These are pieces of code "bound" to a specific model.



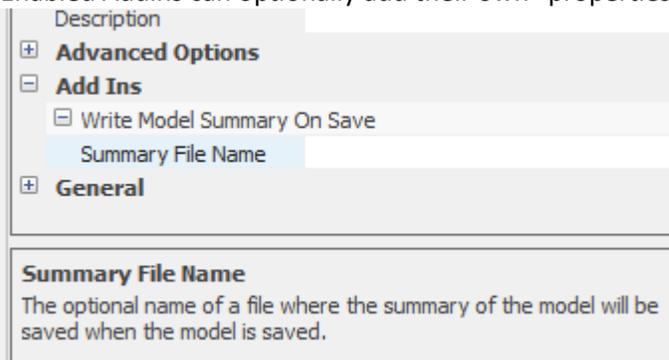
- The place a user starts to implement them is the `SimioAPI.Extensions.IModelHelperAddIn` interface.
- They get loaded with the model and unloaded when the model is closed.
- They have access to the full IModel design time API.
- They can subscribe to events on the model (right now only save).
- They can “augment” the model’s property display and put their own properties in there as well.
- They specify an ‘environment’ under which they are valid to run (right now the ONLY selection is ‘Desktop’).

A normal Simio install will have no model helpers installed along with it. By selecting something in the drop list the user enables (or disables) that Add-In for the active model.

Once ANY add-ins have been enabled, that will disable Undo/Redo. Simio will make sure you know this:



Enabled Addins can optionally add their own “properties” alongside the model’s properties:



Simio Release 10 – Sprint 180/181 – November 28, 2018

This sprint, we have completed our latest multi-sprint round of enhancements related to modeling supply chains -- specifically modeling materials and inventory. Our final set of enhancements provide enhanced inventory control and extensive statistics availability as well as extended support for backorders. We have added an example to show many of these new supply chain features.

We have also dramatically simplified the Table-related ribbons and made them easier to use. We have also added a feature to support more flexible object and element names and added support for the new Professional and RPS Editions.

Material and Inventory - Backorder Policy

Both the Material and Inventory elements have been enhanced to include properties for Allow Backorder Policy options. If a Material element has its *Location Based Inventory* set to 'False', the Allow Backorder Policy and related properties will be available for the material itself. For those materials that are location-based inventory, the *Allow Backorder Policy* information would be specified with the Inventory element associated with the material.

Backorder policies provide an option to always allow backorders, never allow backorders, or conditionally or probabilistically balk at being backordered. Functions for balked and renege backorders are described below. The Inventory and Material Backorders page within Simio help provides detailed information on this feature.



Material and Inventory Stockout Functions and Statistics

A *stockout* is when there is no inventory on hand to fulfill demand. A frequently used service measure is a desired probability of no stockout occurrences during a defined risk period, a criterion sometimes referred to as the *ready rate*. For example, a target service level might be a 95% ready rate.

In Simio, time-dependent statistics on the stockout frequency for each inventory is now automatically collected and reported.

Material element functions include:

Stockout - Provides functions for accessing time-dependent statistics on the material's stockout frequency.

Stockout.TotalTime - Returns the total time that no material was present in the system.

Stockout.PercentTime - Returns the percent time that no material was present in the system.

Stockout.NumberOccurrences - Returns the number of occurrences where no material was present in the system.

Stockout.AverageTime - Returns the average time of the occurrences where no material was present in the system.

Inventory element functions include:

Stockout - Provides functions for accessing time-dependent statistics on the inventory's stockout frequency.

Stockout.TotalTime - Returns the total time that no material was present in the inventory.

Stockout.PercentTime - Returns the percent time that no material was present in the inventory.

Stockout.NumberOccurrences - Returns the number of occurrences where no material was present in the inventory.

Stockout.AverageTime - Returns the average time of the occurrences where no material was present in the inventory.

Automatic Statistics display of these functions will be shown under the Content Category: TimeStockout Data Item section under the Material or Inventory Data Source:

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
Model	Model	Inventory1	AllocationQueue	TimeWaiting	Average (Hours)	0.0125
					Maximum (Hours)	0.2500
					Minimum (Hours)	0.0000
		Content	QuantityConsumed	Total	20.0000	
				QuantityInStock	Average	6.7500
					Maximum	10.0000
		QuantityProduced	Minimum	0.0000		
			Total	20.0000		
		TimeStockout	Average (Hours)	0.5000		
			Occurrences	2.0000		
			Percent	10.0000		
			Total (Hours)	1.0000		

Material and Inventory Fill Rate / Demand / Baked / Reneged Functions and Statistics

Fill rate is the fraction of demand that is immediately satisfied from inventory on hand (no balking or backordering). It is one of the most frequently used service measures in practice to evaluate an inventory's ability to meet demand. For example, a target service level might be a 95% fill rate.

In Simio, two types of fill rate statistics for each inventory will be automatically collected and reported:

Quantity Fill Rate: The percentage of the total quantity demanded that was immediately satisfied from inventory on hand. Sometimes referred to as *unit fill rate* or *case fill rate* in industry.

Order Fill Rate: The percentage of material consumption requests (orders) whose full quantity demanded was immediately satisfied from inventory on hand.

Table 1 shows an example of calculating quantity fill rate versus order fill rate.

Order	Quantity Demanded	Quantity Immediately Consumed	Order Fill Rate Count
-------	-------------------	-------------------------------	-----------------------

1	10	10	1
2	50	40	0
3	30	0	0
4	60	60	1
Total	150	110	2
Quantity Fill Rate = 110/150 = 73.33%			
Order Fill Rate = 2/4 = 50%			

Table 1: Calculating Quantity Fill Rate vs. Order Fill Rate

Balking and renegeing functions are also provided and are based on the *Allow Reorder Policy* properties as described above.

Material element functions include:

Demand - Provides functions for accessing additional statistics on the material's demand.

Demand.NumberOrders - Returns the total number of material consumption requests (i.e., orders) received during the simulation run.

Demand.QuantityDemanded - Returns the total quantity that was demanded to satisfy the material consumption requests received during the simulation run.

Demand.OrderFillRate - Returns the percentage of material consumption requests (i.e., orders) whose full quantity demanded was immediately satisfied from inventory on hand (no balking or backordering).

Demand.QuantityFillRate - Returns the percentage of the total quantity demanded that was immediately satisfied from inventory on hand (no balking or backordering).

Demand.NumberBalked - Returns the total number of material consumption requests that balked at being backordered in an allocation queue).

Demand.QuantityBalked - Returns the total demand quantity cancelled due to balked material consumption requests.

Demand.NumberReneged - Returns the total number of material consumption requests that abandoned waiting as a backorder in an allocation queue.

Demand.QuantityReneged - Returns the total demand quantity cancelled due to renegeed material consumption requests.

Inventory element functions include:

Demand - Provides functions for accessing additional statistics on the inventory's demand.

Demand.NumberOrders - Returns the total number of material consumption requests (i.e., orders) received during the simulation run.

Demand.QuantityDemanded - Returns the total quantity that was demanded to satisfy the material consumption requests received during the simulation run.

Demand.OrderFillRate - Returns the percentage of material consumption requests (i.e., orders) whose full quantity demanded was immediately satisfied from inventory on hand (no balking or backordering).

Demand.QuantityFillRate - Returns the percentage of the total quantity demanded that was immediately satisfied from inventory on hand (no balking or backordering).

Demand.NumberBalked - Returns the total number of material consumption requests that balked at being backordered in the inventory's allocation queue).

Demand.QuantityBalked - Returns the total demand quantity cancelled due to balked material consumption requests.

Demand.NumberReneged - Returns the total number of material consumption requests that abandoned waiting as a backorder in the inventory's allocation queue.

Demand.QuantityReneged - Returns the total demand quantity cancelled due to renege material consumption requests.

Automatic Statistics display of these functions will be shown under the Demand Category under the Material or Inventory Data Source:

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
Model	Model	Inventory 1	AllocationQueue	NumberWaiting	Average	0.8874
					Maximum	1.0000
					Minimum	0.0000
				TimeWaiting	Average (Hours)	1.7778
					Maximum (Hours)	2.0000
					Minimum (Hours)	0.0000
			Content	QuantityConsumed	Total	90.0000
				QuantityInStock	Average	4.5003
					Maximum	15.0000
					Minimum	0.0000
				QuantityProduced	Total	85.0000
				TimeStockout	Average (Hours)	2.0000
					Occurrences	1.0000
					Percent	10.0000
			Total (Hours)		2.0000	
			Demand	NumberOrders	Total	10.0000
				OrderFillRate	Percent	90.0000
				QuantityBackordered	Average	8.8745
					Maximum	10.0000
					Minimum	0.0000
QuantityDemanded	Total	100.0000				
QuantityFillRate	Percent	90.0000				

MultiEchelonSupplyChain – New Example

Adapted from Altiok’s and Melamed’s (2007) book, “Simulation modeling and analysis with Arena,” the Multi-Echelon Supply Chain example from chapter 12.3 has been re-created using Simio. The example has been implemented using data-created and data-driven approaches, where most of the information is stored in tables.

Using Material and Inventory Elements is an alternative to storing and maintaining objects in the system while not in use. The Material and Inventory Elements are logical representations, that can create objects (i.e. entities) when needed, as opposed to having entities idly in the model.

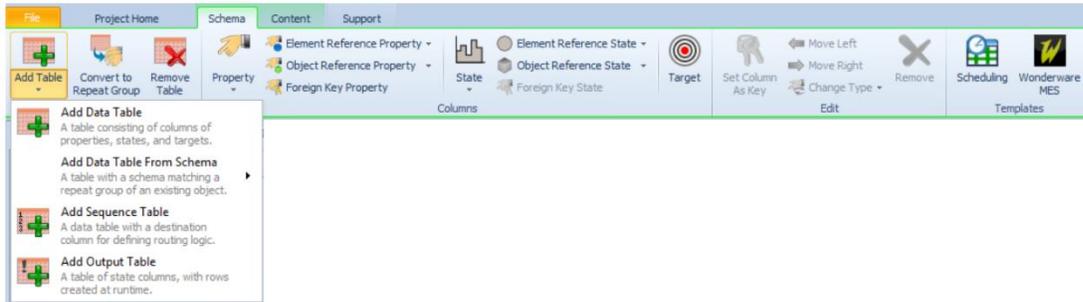
The supply chain is comprised of four locations and five inventories: supplier (raw material), plant (with input buffer for raw material and output buffer for finished products), distribution center (finished product), and retailer (finished product). The supply chain produces one product. For each inventory, except the retailer, backordering is permitted.

At the supplier location, an infinite quantity of raw material is available, which feeds the raw material inventory at the plant location. At the plant, the raw material is transformed into the finished product. The finished product inventory at the plant then supplies products to the inventory at the distribution center, which then supplies the inventory to the retailer location. End-customer orders occur at the retailer location, where a process with similar behavior to a Source is executed.

New Schema/Content Ribbons to Replace Table Ribbon

Within a model’s Data ribbon / Tables panel, the Data window will now include both a Schema and Content ribbon (replacing the Table ribbon and States/Targets ribbons that was shown for some Simio

versions). When you first navigate to Data/Tables, if no tables exist yet, Simio defaults to the Schema ribbon. Otherwise, the Content ribbon is shown.

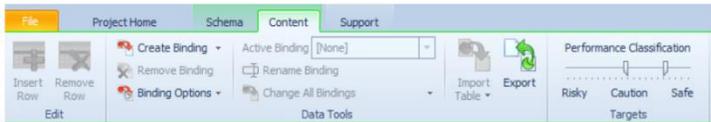


Within the Schema ribbon (above), users can add any type of table, including Data Tables, Sequence Tables and Output Tables (available in Simio Professional and RPS). All users will have the option to add standard Property columns, as well as Element Reference, Object Reference and Foreign Key properties.

The Schema ribbon now contains the contents of the old States ribbon, which will allow users to add standard State columns, as well as Element Reference, Object Reference and Foreign Key states (available in Simio Professional and RPS).

The Schema ribbon also contains a Target (from the old Targets ribbon). Targets are used within the Simio RPS Edition.

The Scheduling and Wonderware MES add-ins now appear on the Schema ribbon, in a new “Templates” group.



The Content ribbon allows users to edit the content of the table, including Binding to data sources, importing and exporting, as well as the adjusting the Target Performance Classification slider.

Simio Professional Edition

Simio Professional Edition provides the same functionality as Simio Design; however, it allows building and distribution of models using the freely available Simio Personal (evaluation) version as a runtime platform. Models built with the Simio Professional Edition will run and generate results with the Simio Personal version. Simio Professional Edition is ideal for consultants that want to deliver a running model to their customer. It also provides the capability to distribute scenarios and replications across computers in a work group. Simio Professional Edition includes Output Tables/States, and Gantt charts for the interactive runs/validation.

Simio RPS Edition

Simio RPS Edition adds a powerful set of patented features to build and execute new models for Risk-based Planning and Scheduling (RPS). This edition provides full scheduling capabilities: custom tailor

reports, graphs, and tables for use by schedulers. Reduce your risk and costs by analyzing your schedules in ways never before possible!

Object and Element Naming Enhancement – RPS Edition

Simio now allows dash and other characters within object and element names. For example, the name of a entity type may be T0-3456, where in prior versions of Simio, the user would have to modify this name slightly to T0_3456 as the '-'dash character was not valid. Similarly, characters such as spaces, \$, #, etc. are now permitted in names. This includes names for objects, such as Servers or elements, such as Materials. Objects and Elements with these additional characters are 'understood' by all versions of Simio, but only can be added/changed within Simio RPS Edition.

Display Category Enhancement- RPS

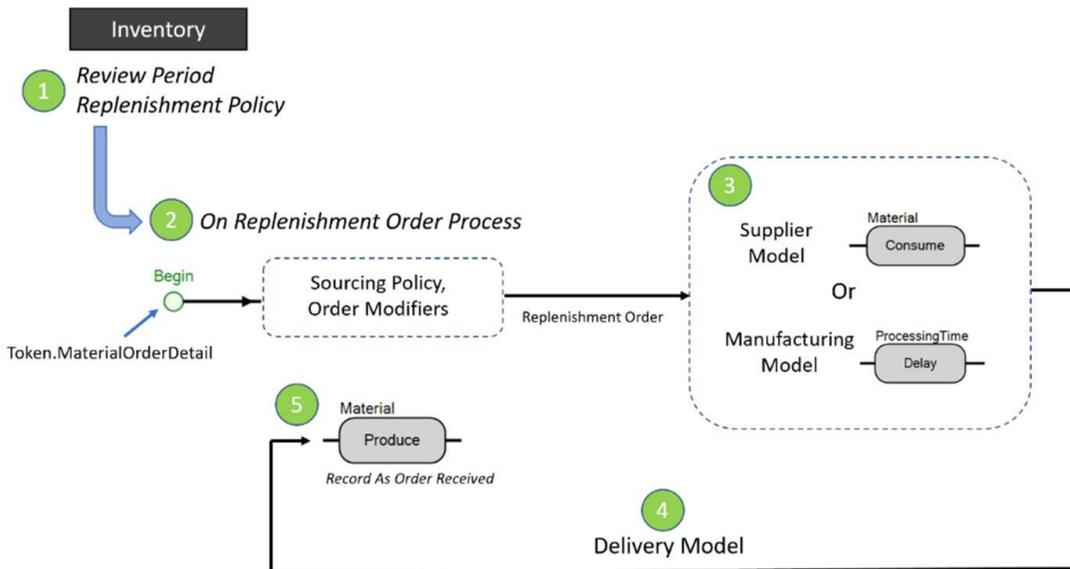
The *Display Category* property on objects that are enabled as resources, such as Server, Worker and Vehicle, can now reference a table value. *Display Category* can be found in the Advanced Options section of properties and is used for hierarchically arranging resources in the Resource Plan Gantt. This enhancement is useful for table generated simulation models, so that the category can be specified in a Resources table.

Simio Release 10 – Sprint 179 – October 15, 2018

This sprint, we are excited about our many Materials and Inventory feature enhancements, including inventory replenishment. The sections below discuss first the general framework and overview of terms and inventory policies, followed by the Simio feature enhancements.

Inventory Replenishment Framework

The graphic below which illustrates the framework that is now available in Simio to model inventory replenishment.



Inventory Position

The *inventory position* is a key input typically used in inventory replenishment decision making. It is equal to inventory-on-hand plus inventory-on-order minus backorders and any other commitments.

Continuous or Periodic Inventory Review

These terms refer to the frequency of inventory review to determine when orders must be placed for replenishment.

A *continuous inventory review* keeps a constant track of the inventory position; as soon as it falls below a pre-determined level (the *reorder point*), a replenishment order is placed. Tracking inventory levels in real-time is typically more expensive to administer but allows for a lower level of safety stock, as the only uncertainty is the magnitude of demand during the delivery period.

Alternatively, a *periodic inventory review* evaluates the inventory position at discrete points in time to determine if a replenishment order needs to be placed. Replenishment decisions can be made only at those points. The time between two review points is called the *review period*. A periodic review system is cheaper to administer compared to continuous review since inventory counts take place only at fixed times, but a higher level of safety stock is typically required to buffer against a longer period of uncertainty in demand.

Replenishment Policies

At the time of an inventory review, a decision strategy often referred to as the *replenishment policy* is used to determine whether replenishment is required and, if so, then by how much.

Some commonly used replenishment policies:

Min/Max Replenishment Policy

This policy is sometimes referred to as the (s, S) policy where s is the reorder point, the 'Min', and S is the order-up-to level, the 'Max'. When the inventory position falls to or below the reorder point, s , then replenishment is required to bring the inventory position to the order-up-to level, S . In other words, if the inventory position is y , and $y \leq s$, then a replenishment order of size $S - y$ is required.

Figures 1 & 2 illustrate the Min/Max replenishment policy using either continuous or periodic inventory review.

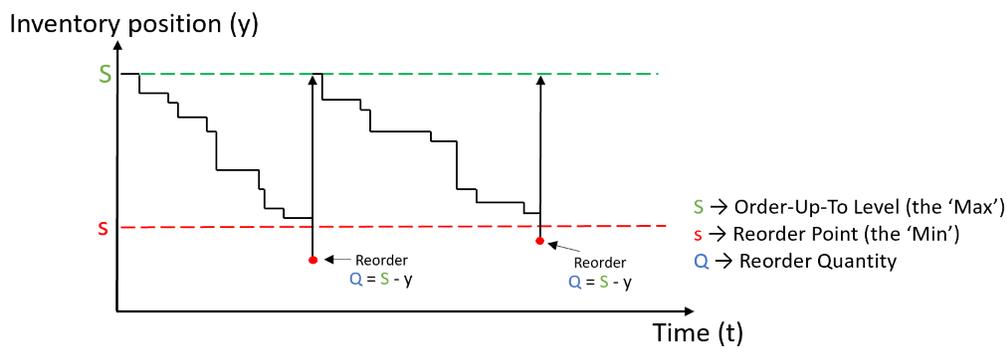


Figure 1 – Continuous Review Min/Max Replenishment Policy

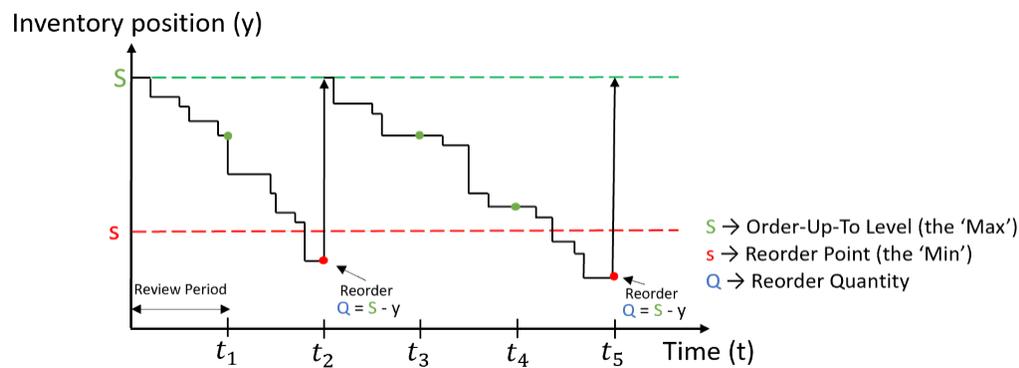


Figure 2 – Periodic Review Min/Max Replenishment Policy

Order-Up-To Replenishment Policy

This policy is sometimes referred to as the *base-stock policy* or "one-for-one" policy. When the inventory position decreases below the order-up-to level, S , then replenishment is required so as to bring the inventory position back to the order-up-to level. In other words, if the inventory position is y , and $y < S$, then a replenishment order of size $S - y$ is required.

Figures 3 & 4 illustrate the Order-Up-To replenishment policy using either continuous or periodic inventory review.

Inventory position (y)

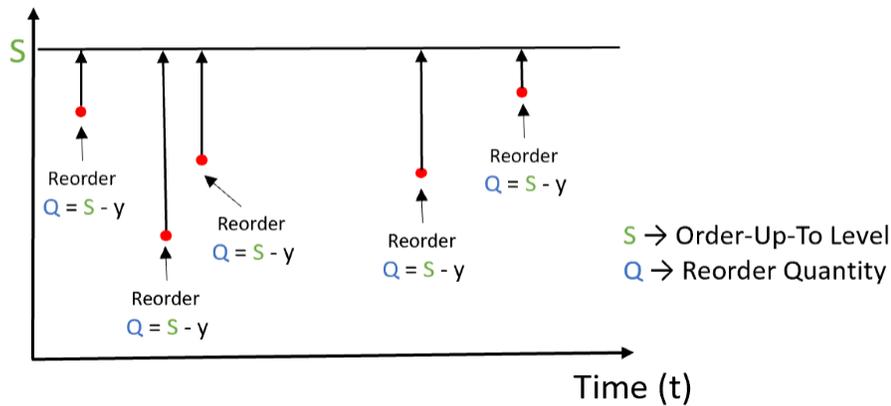


Figure 3 - Continuous Review Order-Up-To Replenishment Policy

Inventory position (y)

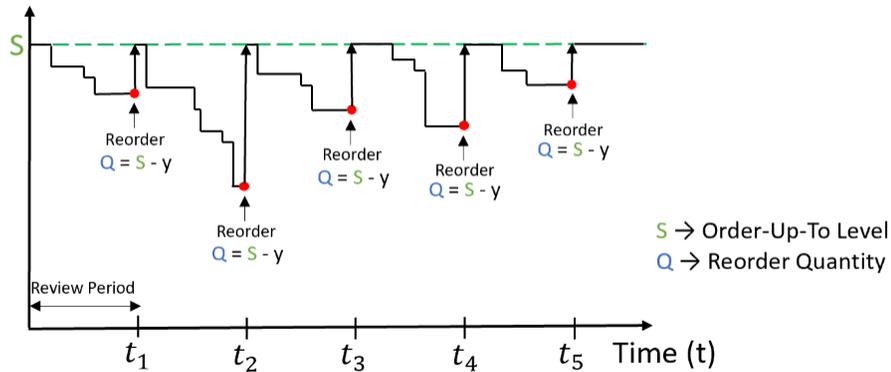


Figure 4 - Periodic Review Order-Up-To Replenishment Policy

Reorder Point/Reorder Qty Replenishment Policy

This policy is sometimes referred to as the (s, nQ) policy where s is the reorder point and Q is a fixed reorder quantity (a fixed lot size). When the inventory position falls to or below the reorder point, s , then replenishment is required to bring the inventory position just above s . The size of the replenishment order is a multiple of the reorder quantity, Q . In other words, if the inventory position is y , and $y \leq s$, then a replenishment order of size nQ is required, where n is the smallest integer such that $y + nQ > s$.

Figures 5 & 6 illustrate the Reorder Point/Reorder Qty replenishment policy using either continuous or periodic inventory review.

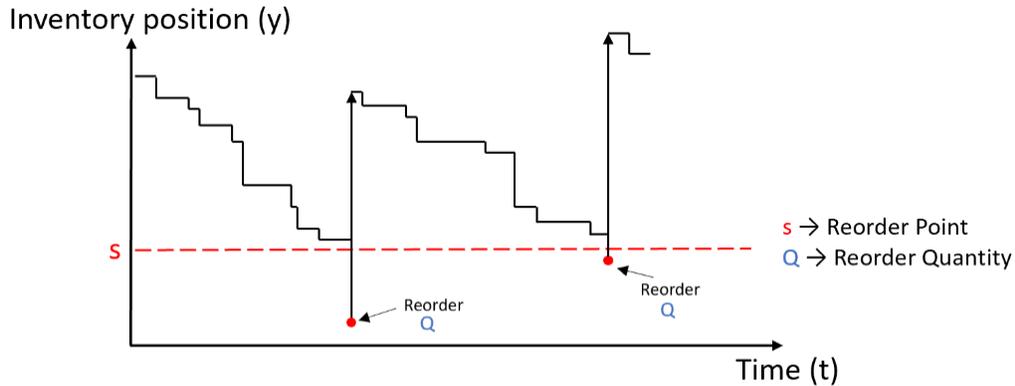


Figure 5 - Continuous Review Reorder Point/Reorder Qty Replenishment Policy

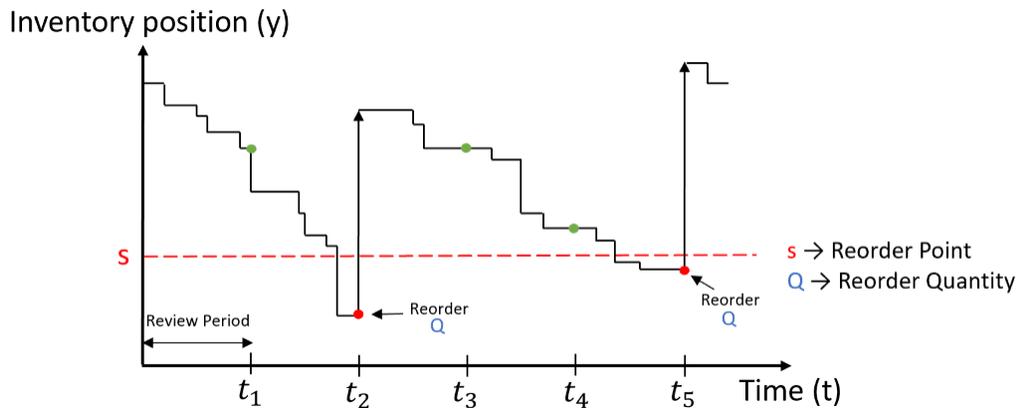
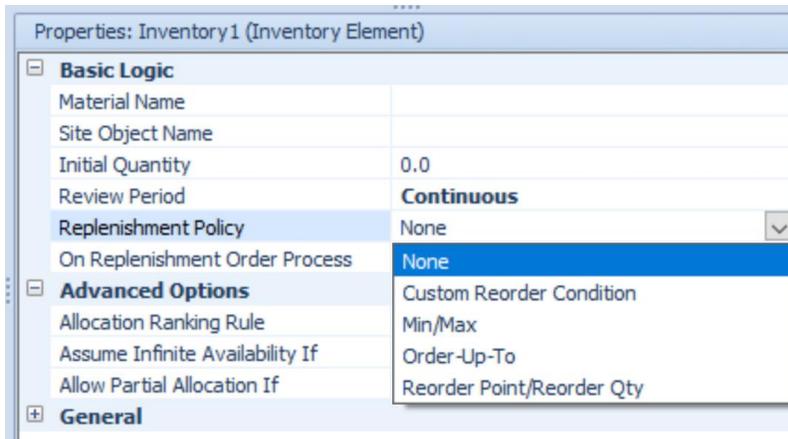


Figure 6 - Periodic Review Reorder Point/Reorder Qty Replenishment Policy

Inventory Element – Replenishment Features

Within the Inventory element, we have added multiple properties to handle inventory replenishment, including *Review Period*, *Replenishment Policy* and *On Replenishment Order Process*. The *Review Period* determines the frequency of inventory review to determine whether a replenishment order is required. This can be 'None', 'Continuous' or 'Timer' (periodic as discussed above). The *Replenishment Policy* property can be specified as 'None', 'Custom Reorder Condition', 'Min/Max', 'Order-Up-To' or 'Reorder Point/Reorder Qty'. Finally, an *On Replenishment Order Process* property can be executed to handle an inventory replenishment order. The initiation of this process automatically increments the inventory's quantity on order. The created token's material order detail reference (new) will provide the detail of the order replenishment.



Material and Inventory Elements – QuantityOnOrder state

For both the Material and Inventory elements, we have added a *QuantityOnOrder* state variable that returns the quantity of the material that has been ordered to replenish inventory in the system but has not yet been received.

If a material is not location-based inventory, this value will return 0. Otherwise, it is automatically assigned the value of the total aggregate quantity on order across all inventory sites and is not user-assignable. The inventory's *QuantityOnOrder* state may also be assigned using the Assign step.

Inventory Element – InventoryPosition function

A new *InventoryPosition* function has been added for Inventory elements that is equal to the inventory's quantity in stock plus quantity on order minus quantity backordered minus quantity reserved (excluding reserved units already backordered).

Consume Step – On Consumed Process

The Consume step now includes an option process to be executed when material is consumed. The created token's material order detail reference will provide the detail of the material consumption. This process will be executed any time a quantity of material is consumed; either a full or partial allocation.

Produce Step – Record As Order Received If Property

The Produce step now includes a *Record As Order Received Condition* property which, if 'True' and the material is location-based inventory, then the inventory's quantity on order is automatically decremented by either the produced quantity or to zero, the latter if the produced quantity is greater than the quantity on order.

Token Function Enhancements

The following new functions will be provided by a process token:

MaterialOrderDetail - Returns a reference to the material order detail associated with the token, if applicable.

MaterialOrderDetail.Material - Reference to the material.

MaterialOrderDetail.SourceInventory - Reference to the inventory that is the source of the material.

MaterialOrderDetail.SourceSiteObject - Reference to the inventory site object that is the source of the material.

MaterialOrderDetail.DestinationInventory - Reference to the inventory that is the destination of the material.

MaterialOrderDetail.DestinationSiteObject - Reference to the inventory site object that is the destination of the material.

MaterialOrderDetail.Quantity - The quantity of the material.

MaterialOrderDetail.LotID - The lot identifier of the material.

Note that If a token has been created to execute an *On Replenishment Order Process* specified for an Inventory element, then the destination in its material order detail will be the inventory site requesting the replenishment. Since built-in features for modelling sourcing policies will not yet be available, the source in the token's material order detail will always be set to nothing.

If a token has been created to execute an *On Consumed Process* specified for an Consume step, then the source in its material order detail will be the inventory site from which the material was consumed. The destination in its material order detail will be copied from the material order detail of the original token, if applicable.

Simio API Extensions Enhancements

This section describes the interfaces that will be provided within the SimioAPI.Extensions namespace to create user-coded replenishment policies.

IReplenishmentPolicyDefinition Interface

string Name { **get**; } - Name of the replenishment policy. May contain any characters.

string Description { **get**; } - Description text for the replenishment policy.

void DefineSchema(IPropertyDefinitions propertyDefinitions) - Guide which uniquely identifies the replenishment policy.

IReplenishmentPolicy CreatePolicy(IPropertyReaders properties) - Defines the property schema for the replenishment policy.

IReplenishmentPolicy CreatePolicy(IPropertyReaders properties) - Creates a new instance of the replenishment policy.

IReplenishmentPolicy Interface

bool CheckPolicy(IReplenishmentPolicyContext context, **out double** orderQuantity) - Checks the replenishment policy to determine whether a replenishment order is required. Returns true if a replenishment order is required, along with the order quantity; otherwise, returns false.

IReplenishmentPolicyContext Interface

double CurrentInventoryPosition { **get**; } - The current inventory position.

Note: When getting the value of a property in the IReplenishmentPolicy.CheckPolicy() method using the IReplenishmentPolicyContext *context* argument, that execution context is going to be the inventory element. Thus, the property value may be a table row reference that has been set for the inventory element.

Simio Release 10 – Sprint 178 – September 24, 2018

This sprint we are continuing to add many new features to our material/inventory functionality. For the next several sprints, we will highlight the areas that have been enhanced with inventory management and materials.

Material and Inventory – Partial Allocation

For a Material element, there is now an *Allow Partial Allocation If* property that is a default optional condition indicating whether an entity may immediately consume only a portion of its requested quantity if the full quantity is not available. If the condition evaluates to true, the entity will consume the partial amount, but will still wait until the entire quantity fulfilled before moving from Consume step.

For an Inventory element, we also added an *Allow Partial Allocation If* property that is an optional condition indicating whether an entity may immediately consume only a portion of its requested quantity if the full quantity is not available. If specified, then this condition will override the *Allow Partial Allocation If* on the Material element associated with the inventory.

Simio Release 10 – Sprint 177 – August 31, 2018

This sprint we have added a new SimBit that gives a simple view of our new inventory features that were added in Sprint 176. We've also updated a SimBit for database writing. Several user issues have been fixed and we've added a user enhancement request of exporting from various table areas to different formats.

SimBits – New and Updated

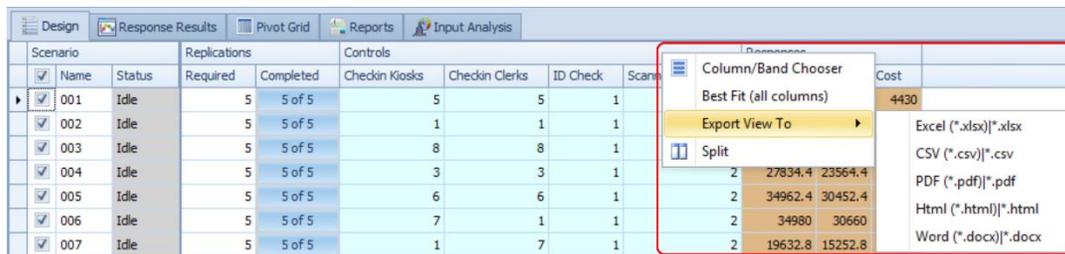
We have added a new SimBit to illustrate the new inventory type features that were added to the previous Sprint 176. Additionally, we have modified the DBReadWrite SimBit to use SQL Server Express.

InventoryAndMaterials – This model illustrates an example where materials are stored in and consumed from inventories at different locations. The simulation model shows material consumption from site-specific inventories.

DBReadWrite – This SimBit has been updated to use SQL Server Express instead of MySQL. This model demonstrates how to read data into Simio and write data from Simio using SQL Server.

Right-Click – Export View To

We have added a new right-click option on the table areas (tables, experiment, etc.) that allows users to export the table view to several different sources, such as Excel, CSV, PDF, Html and Word.



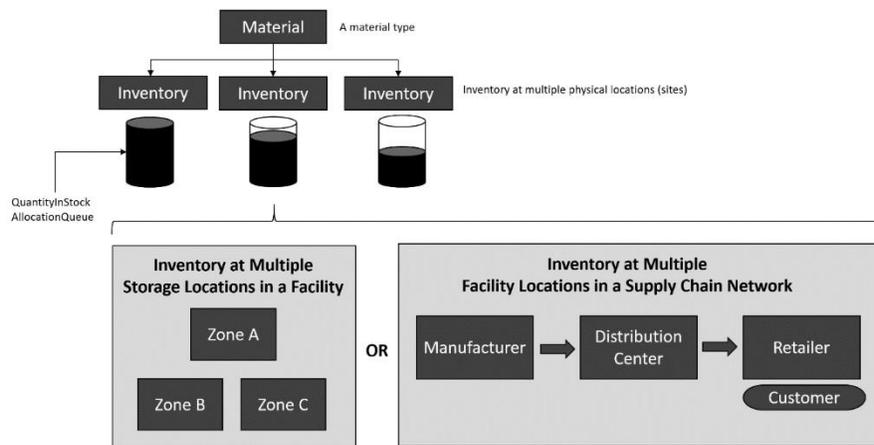
Scenario	Name	Status	Replications	Completed	Checkin Kiosks	Checkin Clerks	ID Check	Scans	Cost
001	001	Idle	5	5 of 5	5	5	1		4430
002	002	Idle	5	5 of 5	1	1	1		
003	003	Idle	5	5 of 5	8	8	1		
004	004	Idle	5	5 of 5	3	3	1		
005	005	Idle	5	5 of 5	6	6	1		
006	006	Idle	5	5 of 5	7	1	1		
007	007	Idle	5	5 of 5	1	7	1		

Simio Release 10 – Sprint 175/176 – August 14, 2018

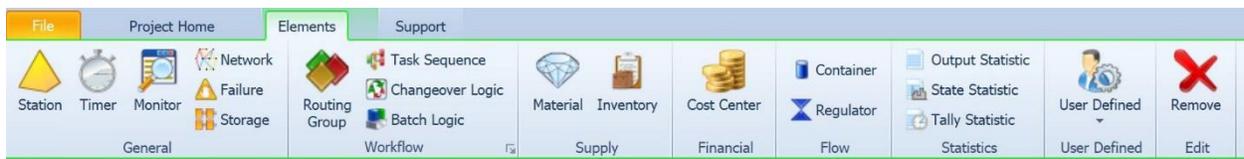
This sprint we have added a new Inventory element that makes it easier for users to model material stock that is stored at multiple physical locations in the system, with each inventory having its own stock level and allocation queue. Now, a fixed object within the Facility window (such as Server or TransferNode) can be designated as an inventory's physical location (inventory site).

New Inventory Element

A new Inventory element may be used to define a storage bucket for holding stock of a specific material. A *Site Object Name* property is provided to designate a fixed object in the Facility window, such as a Server or TransferNode, as the inventory's physical location.

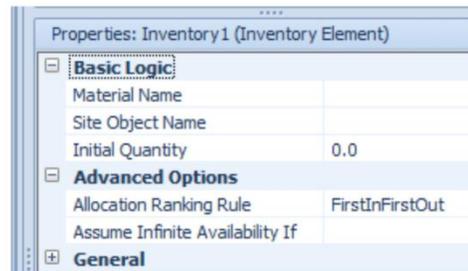


The Inventory element is now available from the Elements ribbon. *Note: Organization of buttons within the Elements ribbon has also changed slightly.



Materials specified within the simulation model may then be associated with the inventory locations.

Name	Object Type
▼ Inventory Elements	
Inventory1	Inventory Element



States and Functions for Inventory Element

The states for the Inventory element include:

QuantityInStock - The current quantity of material present in the inventory.

AllocationQueue - Accesses the inventory's queue of material consumption requests.

The functions for the Inventory element include:

AverageQuantityInStock - Returns the average quantity of material present in the inventory during the simulation run.

MinimumQuantityInStock - Returns the minimum quantity of material present in the inventory during the simulation run.

MaximumQuantityInStock - Returns the maximum quantity of material present in the inventory during the simulation run.

QuantityConsumed - Returns the total quantity of material removed from the inventory during the simulation run.

QuantityProduced - Returns the total quantity of material added to the inventory during the simulation run.

QuantityReserved - Returns the current quantity of material from the inventory that is reserved.

QuantityReserved (owner) - Returns the current quantity of material from the inventory that is reserved for use by a specified owner object.

QuantityBackordered - Returns the current quantity of material required to satisfy all material consumption requests waiting in the inventory's allocation queue.

AverageQuantityBackordered - Returns the inventory's average quantity backordered during the simulation run.

MinimumQuantityBackordered - Returns the inventory's minimum quantity backordered during the simulation run.

MaximumQuantityBackordered - Returns the inventory's maximum quantity backordered during the simulation run.

UsageCostCharged - Returns the total cost that has been charged to consumers of material from the inventory.

Material Element Enhancements

The Material element has been enhanced to include the *Location Based Inventory* property (default set to 'False'). When the material is location based, the *Initial Quantities* and *Allocation Ranking Rule* will be specified on the associated Inventory element.

New Material Element Functions

With the Material element enhancements, we have added the following new functions for materials:

Inventory (site) - Returns a reference to the inventory holding this material at the specified site (a fixed object). If there is no such inventory defined, then the Nothing keyword is returned.

QuantityBackordered - Returns the current quantity of material required to satisfy all material consumption requests waiting in any allocation queues.

AverageQuantityBackordered - Returns the material's average quantity backordered during the simulation run.

MinimumQuantityBackordered - Returns the material's minimum quantity backordered during the simulation run.

MaximumQuantityBackordered - Returns the material's maximum quantity backordered during the simulation run.

UsageCostCharged – Returns the total cost that has been charged to consumers of the material.

** NOTE: The MaterialCostCharged function is now deprecated, use the UsageCostCharged function instead.

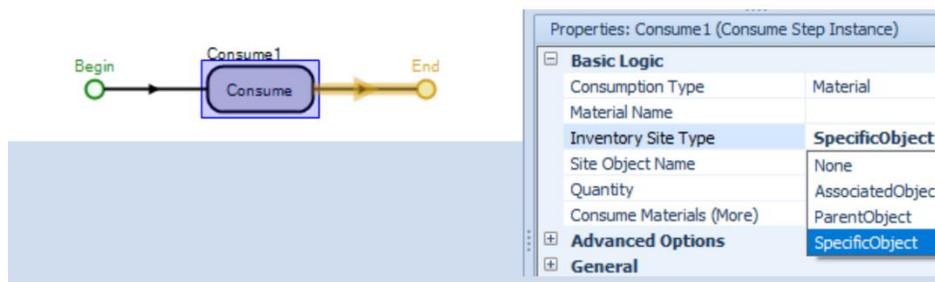
Fixed Object Function Added

When a fixed object, such as a Server, Combiner, Separator or TransferNode, is specified as the *Site Object Name* of an Inventory element, the function Inventory (material) can be used.

Inventory (material) – Returns a reference to the inventory holding the specified material at this fixed object. If there is no such inventory defined, then the Nothing keyword is returned.

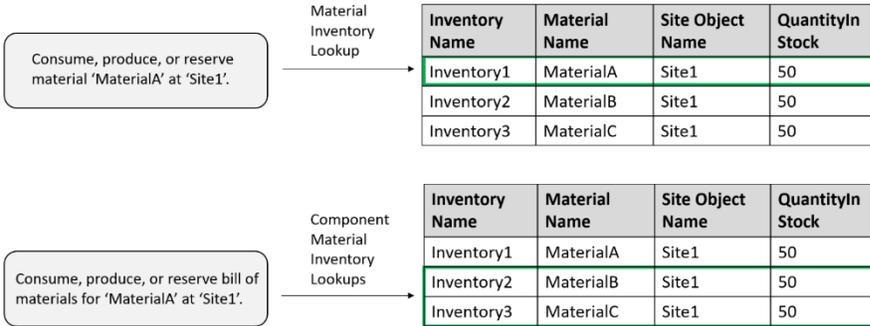
Consume, Produce, Reserve, UnReserve and Route Step Enhancements

The Consume, Produce, Reserve, UnReserve and Route steps now include an *Inventory Site Type* property which can be specified as 'None' (default), 'AssociatedObject', 'ParentObject' or 'SpecificObject'. Additionally, the *Site Object Name* property will then be visible if the *Inventory Site Type* is 'SpecificObject'. This allows these steps to then work with the Inventory element features.



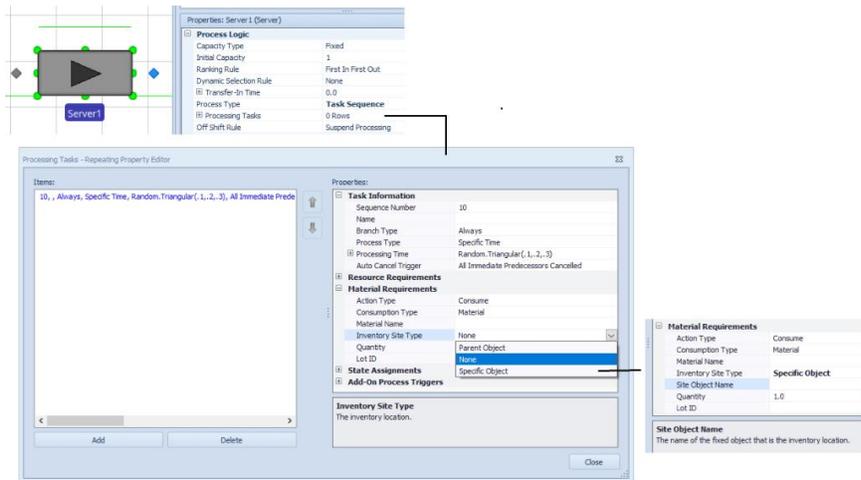
If consuming, producing, reserving or unreserving a single material that is location-based inventory, then Simio will look up the Inventory element holding the specified material at the specified inventory site. If producing a bill of materials, then Simio will look up the inventory elements holding the component materials at the specified inventory site. This holds true for routing with required materials as well.

MaterialA
 Bill Of Materials
 2 Units of MaterialB
 4 Units of MaterialC

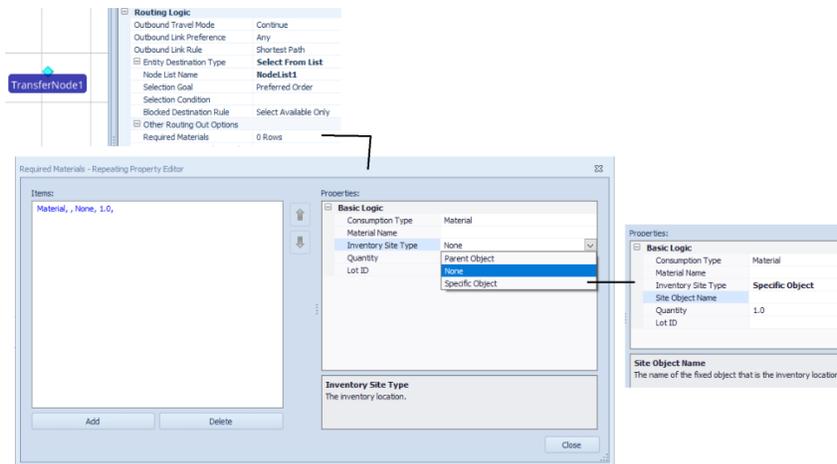


Standard Library Enhancements - Server, Combiner, Separator and TransferNode

The Server, Combiner, Separator objects have been modified to have *Inventory Site Type* and *Site Object Name* properties within the Task Sequences, Required Materials section of properties. This will allow users to specify, on a task basis, the inventory site area (if any) from which to consume or produce materials.



Additionally, the TransferNode object has been enhanced to allow required materials before routing out to include material that is location-based inventory.



Material Usage Log Enhancement

If logging material consumption and production, then the **Material Usage Log** now includes a new *Site Id* field and the existing *Stock Level* field will now be the site location quantity in stock.

Resource Usage Log / Task Log Enhancement

We have added two properties to the Custom Column fields within the Resource Usage Log and the Task Log. The 'Add Column' button on the Log ribbon is used to add custom columns to the logs. This is typically done to provide the user with 'Additional Details' information on a tooltip or to categorize the resources or tasks. However, there are some instances when the extra column is used to provide information on the Gantt (border color, for example) but the user may not wish to have it shown on the hover tooltip and/or categorization fields dropdowns. The *Show in Gantt Tooltips* and *Show in Gantt Dropdowns* properties have been added to allow the user to turn off ('False') the display of the custom column in various places.

OptQuest – Minimum Replications

Within Simio experiments using OptQuest, users can now specify the minimum replications (*Min Replications* property) to 1. Previously, this value was required to be 5 or more.

Ability to View Product Version for Model

We have added the ability to view the product version and date with which the model was saved. Users can right click on the Project Name within the Navigation window to see the Project Properties. The *Date Saved* and *Saved In Version* fields are now available, as seen below in red. This was a user requested feature.

Navigation: TestVersionNumber

- TestVersionNumber
 - Mod New Model
 - Mod New Entity
 - Rename Project
 - Project Properties

Properties: TestVersionNumber (Project)

Data	
CSV Separator	
Disable Grid Dat...	False
Advanced Options	
Save Project As...	True
General	
Project Name	TestVersionNumber
Icon	<input type="checkbox"/> (none)
Date Saved	8/13/2018 2:41:48 PM
Saved In Version	10.175.17067.0

Simio Release 10 – Sprint 174– June 28, 2018

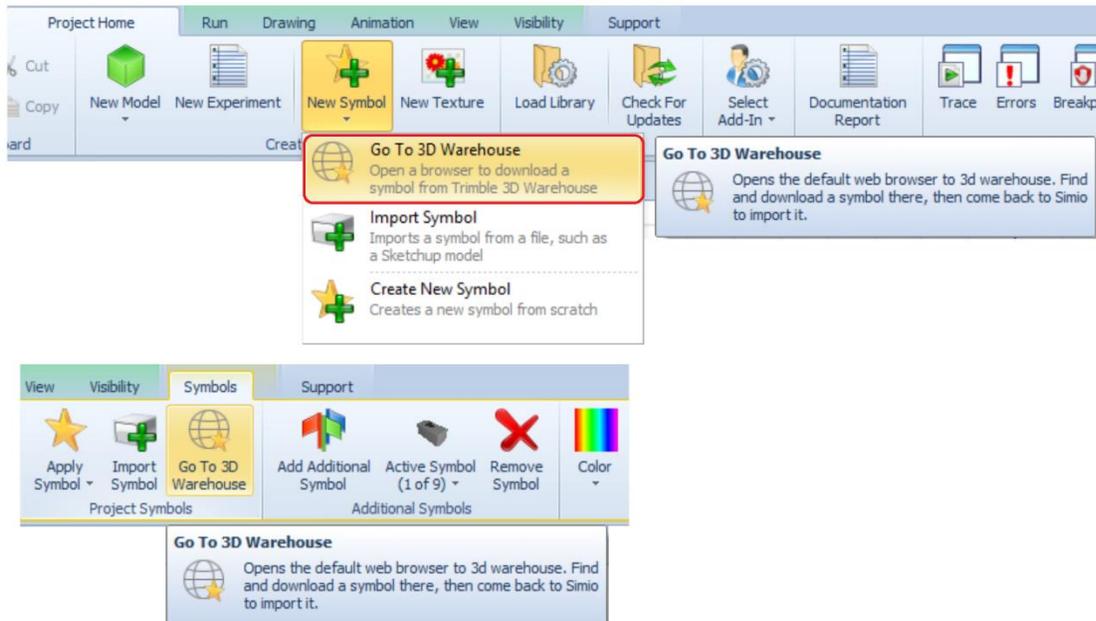
This Simio sprint 174 is our summer public release. In addition to the many features we have added since the last public release sprint 165, this sprint we have modified the 3D Trimble Warehouse interface, enhanced the Gantt charts to include visualization of task constraints in the main Gantts, and added an option to destroy the entity coming from a TransferNode. We have also expanded our Simio Personal Edition software to allow for small scheduling models and added the first phase of a Visio interface on our Simio Users Forum - Shared Items.

Simio Personal Edition – Expanded!

We have expanded the Simio Personal Edition (free for download) to include Simio Enterprise Edition functionality (Planning tab). Users will be able to download the personal edition software to run any of the Simio examples (including scheduling examples) provided with the software. In addition to building simulation models within the software, small scheduling models can be built to demonstrate the scheduling capability to potential users evaluating Simio.

Update to 3D Trimble Warehouse Importing

Based on changes that Trimble 3D Warehouse has recently made, we have had to modify our interface to that application. Now, when a symbol will be added or modified, there is an option to Go To 3D Warehouse (prior was named Download Symbol). This button will put the user in a web browser for 3D Warehouse where they can search for symbols and download to a Sketchup file. Sketchup files can then be imported using the Import Symbol button.



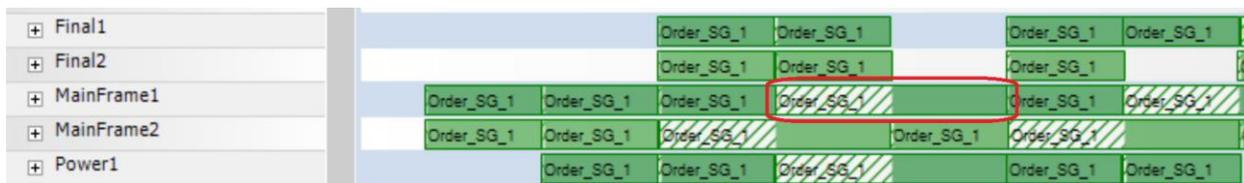
Support for Sketchup 2018

We have now added support for importing Sketchup 2018 files. Within the Go To 3D Warehouse button discussed above, the user has the option of which Sketchup version of the software the symbol will be downloaded.

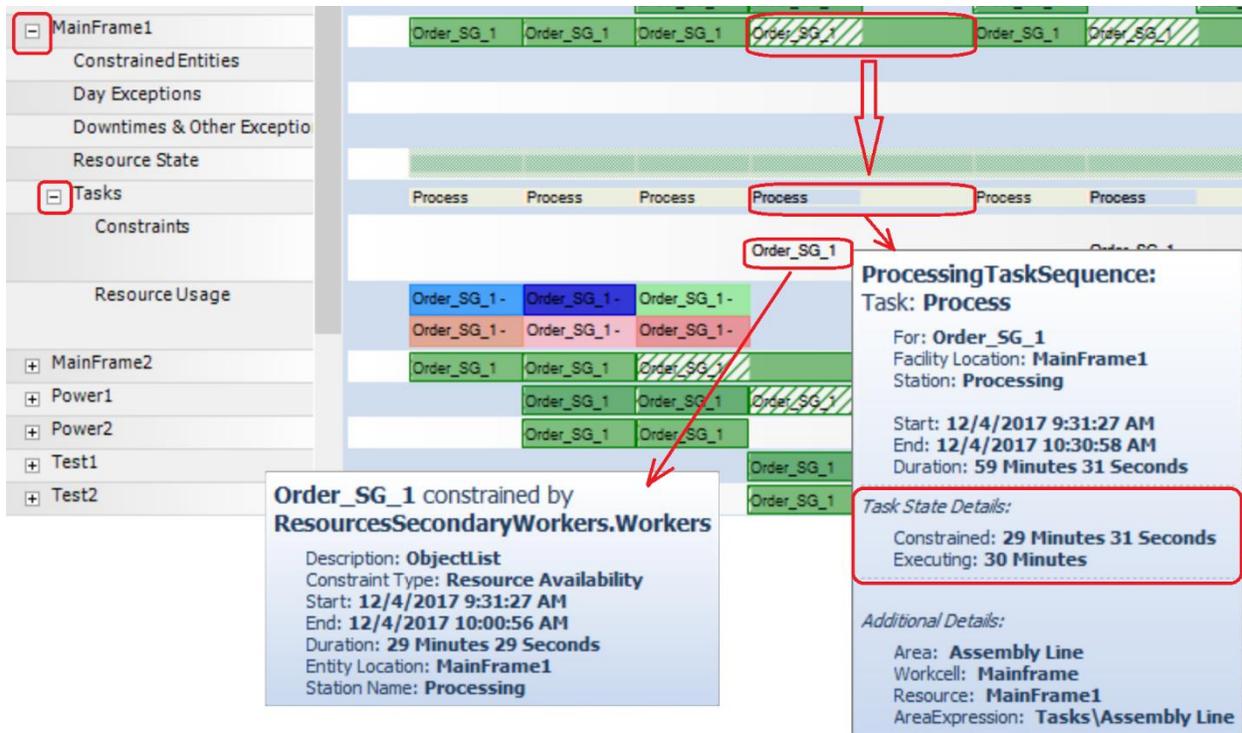
Gantt Enhancements – Enterprise Edition

We have enhanced the Resource Plan and Entity Flow Gantts to include hatching on the main bars of the Gantt when a resource/entity is constrained. In prior sprints, the main bar of a resource/entity was solid color to show when the resource was seized (Resource Plan Gantt) or when the entity was processing (Entity Flow Gantt). While this solid color bar is fine when there are no constraints (secondary resources, materials, offshift times, etc.), it did not show any indication of constraint issues unless the + expansion button was used to expand to those various constraints.

Users will now see, in the main Gantt bars, a hatch pattern indicating a constraint. The example below shows Order_SG1 is constrained for *some reason* for the first half of its time range on MainFrame1 (circled in red).



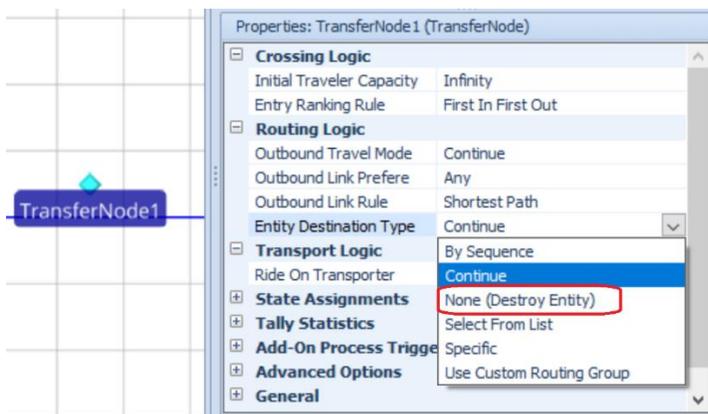
By using the + to open up the MainFrame1 resource, the Process task is shown as constrained. Hovering over the “Process” bar will show the task state details and time of constrained / executing states. Using the + to open the Tasks will show the constraints for the task (resources/materials). Hovering over the constraint will show the specific details for the entity, in this case resource availability of a secondary resource.



Double-clicking on the particular entity within the Resource Plan Gantt will take the user to the corresponding Entity Gantt view.

TransferNode – Entity Destination Type

We have added an option to the *Entity Destination Type* property within the TransferNode object (and all fixed objects that contain TransferNode). The user now has the option, upon leaving a TransferNode, to destroy the entity. The new option is 'None (Destroy Entity)'.



New and Modified SimBits

We have added several new SimBits (and modified one) to demonstrate the use of dashboards, combining specific entity types and quantities and loopback branches within task sequences.

DashboardsForSchedulingExamples – This model demonstrates how to show results in graphical format using Dashboards with an Enterprise version. The model is based on the Scheduling Discrete Part Production Example.

DashboardReportTallies – This model has been modified to include logging for all part type tallies and includes all entity tallies within the dashboard. The dashboard has also been enhanced to include three different sections of displaying information.

CombineMultiplePartsTaskSequence – This model demonstrates how to batch different entities types together for subsequent processing using the Combiner’s Task Sequence. Each task batches a different entity type / quantity and uses a string value to match the member to the task.

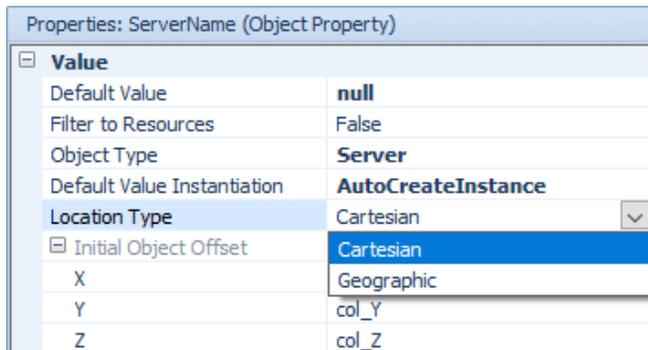
ServersUsingTaskSequenceWithDataTables_LoopbackBranches – This project contains two models in which the operations performed at servers are task sequences, and there are conditional or probabilistic loopback branches in the task workflow. All operation data is defined in a set of relational data tables.

Updated Scheduling Examples

We have updated both the SchedulingBatchBeverageProduction and the SchedulingBicycleAssembly examples to use the auto-create objects functionality within the Resources tables. Note that custom objects are used in conjunction with Simio objects to build these models using the Resource Name column and x/z locations within the Resources table. Documentation and screenshots have also been updated with these examples.

Auto Creation of Objects – Enhancement

When using a table to auto-create object instances, users can now specify each object’s *Initial Object Offset* coordinates using Geographic values. Object Property has a new *Location Type* option to indicate if it uses Cartesian (x, y, z) or Geographic (latitude, longitude, elevation) coordinates.



Selecting the default value of ‘Cartesian’ allows users to enter expressions for X, Y, and Z, which is existing behavior. Selecting the new value of ‘Geographic’ allows users to enter expressions for Latitude (-90.0 to +90.0), Longitude (-180.0 to +180.0), and Elevation (in meters, which is just like the existing Y Cartesian coordinate, but available here for completeness). Like the X, Y and Z values, the new Latitude, Longitude and Elevation values can optionally be specified as table columns and will take their values from the corresponding row in the table.

Properties: ServerName (Object Property)	
<input type="checkbox"/> Value	
Default Value	null
Filter to Resources	False
Object Type	Server
Default Value Instantiation	AutoCreateInstance
Location Type	Geographic <input type="checkbox"/>
<input type="checkbox"/> Initial Object Offset	col_Lat,col_Lon,col_Elev
Latitude	col_Lat
Longitude	col_Lon
Elevation	col_Elev

Object instances created using Geographic coordinates will remain at those coordinates even if the Facility window's Map Location origin is changed.

Visio Interface Release

Simio is releasing the first phase of a Simio-Visio interface that permits Visio drawings to be imported into Simio. This new capability can be found in the Shared Items forum of the Simio Insiders Forum.

This first phase provides two ways to input Intelligent Objects, Links, and Vertices from Visio into the Simio Facility view:

1. Use Simio-Provided Visio Stencils to add Intelligent Objects and their Links, or
2. Add Shape-Data (Visio properties) to existing Visio drawings to identify the Shapes and Connectors destined for Simio.

Additional features are:

- A Visio license is not required by the Add-In (only a Visio .vsdx file)
- No COM or .NET Interop or 3rd party software is required.

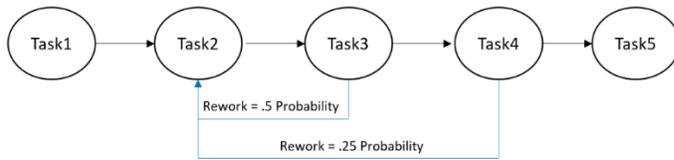
Later phases will permit the import of Visio data for Simio Processes and Tasks.

Simio Release 10 – Sprint 172/173 – June 12, 2018

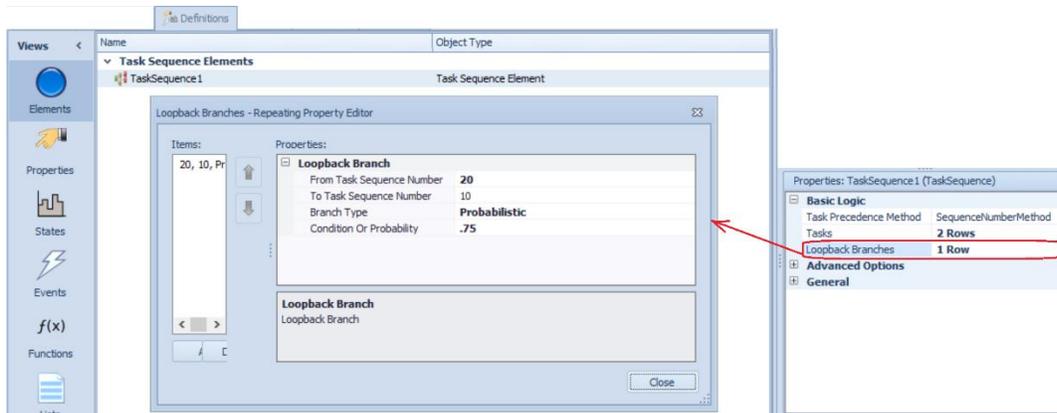
This sprint, we have made several user-requested enhancements to Task Sequences include loopback branches and state assignments within the tasks. We have also fixed some breakpoint issues and changed the way that breakpoints are handled. The Create Object From This feature (added in Sprint 165) has been enhanced as well to include an updating option. These and several other enhancements were requested or discussed last month at our User Group Meeting.

Task Sequences Element Enhancement – Loopback Branches

We have enhanced the Task Sequences element to include a repeating property editor of loopback branches. This will allow for users to include conditional and/or probabilistic looping within a Task Sequence. In the below example, after Task3, 50% of the entities will loop back to Task2 and continue processing through the task sequence again. After Task4, 25% of the entities will loop back to Task2 and continue processing through the task sequence again.

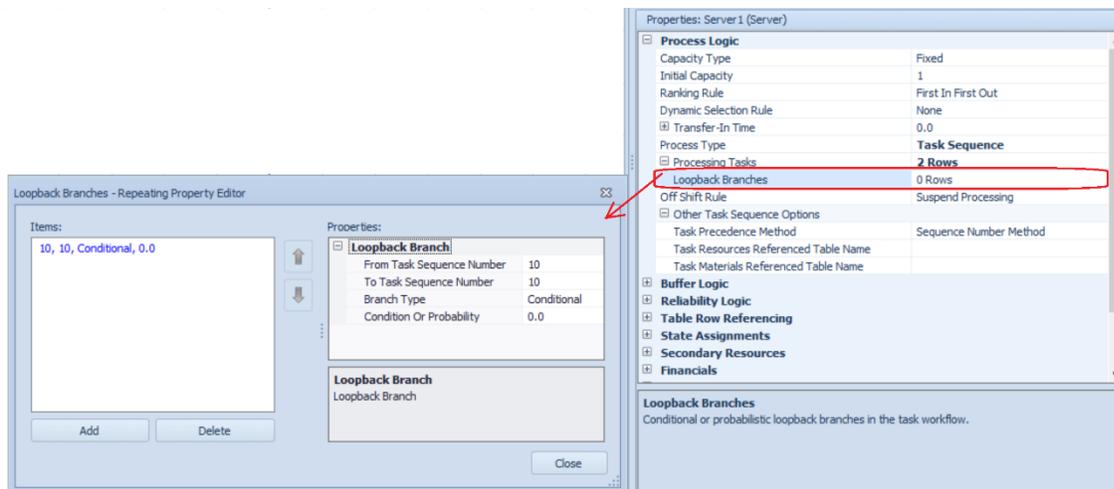


The below diagram shows the required information required for 'SequenceNumberMethod' type sequences. The *From Task ID Number* and *To Task ID Number* properties are used for a *Task Precedence Method* of 'ImmediatePredecessorsMethod' or 'ImmediateSuccessorsMethod'. Multiple loopback branches may be specified from the same task.



Server, Combiner and Separator – Loopback Branches Enhancements

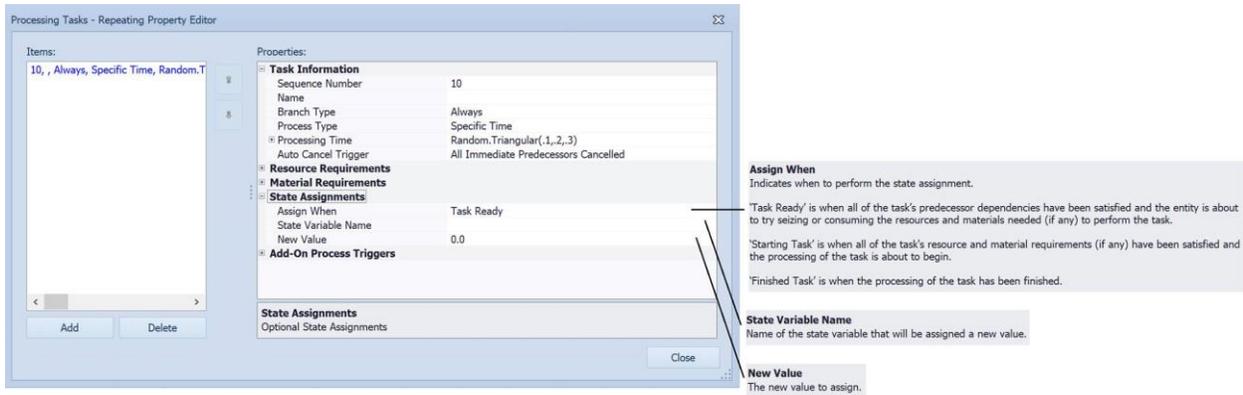
With the addition of the loopback branches to the Task Sequence element (noted above), these same enhancements have been made to the Server, Combiner and Separator. All these objects currently include the Task Sequence option for specifying processing information. If the object's *Processing Type* is set to 'Task Sequence' and there are one or more entries in the *Processing Tasks* repeating editor, the Loopback Branches option will be shown.



Server, Combiner, and Separator – Task Sequence Process Type Enhancements

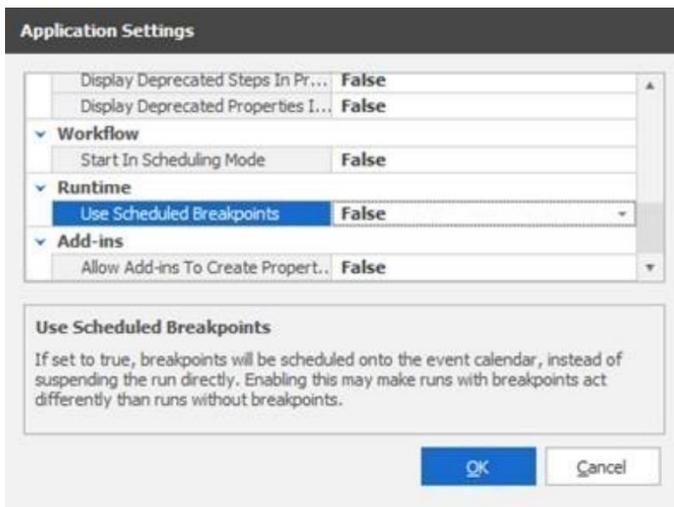
A State Assignments property category is now available for an individual processing task within the Task Sequence processing of a Server, Combiner or Separator. This will allow users to make changes to state

variables within the repeatable property editor instead of through an add-on process. State assignments can be made when the task is ready to begin (not yet seizing resources or consuming materials), starting the task (resources and materials available, task delay about to start) or completed.



Breakpoints Enhancement

There is a new setting in the File -> Settings dialog, defaulted to False (opting everyone into “perfect breakpoints”). We have had a several instances where user’s models behaved slightly differently if a breakpoint was set than if it wasn’t set and have therefore added this feature.



Trace Enhancement – Include State Variable's Previous Value in Trace Message for Assign Step

We have updated the trace action statement for Assign steps to include the original state value as well as the new state value. This was a feature requested by customers for easier viewing and debugging in the Trace window.



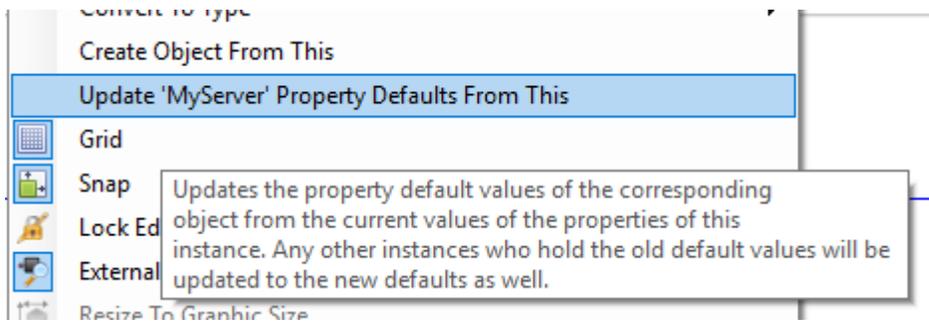
Task State Log – Entity Id and Entity

We have enhanced the Task State Log (originally added sprint 170) to include two new columns including the Entity Id and Entity. These fields were added to provide better dashboard support between various logs.

Task Id	State	Entity Id	Entity	Start Time	End Time	Duration (Hours)
1	Executing	DefaultEntity.39	Order05	12/1/2017 8:00:05 AM	12/1/2017 9:30:05 AM	1.5
3	Executing	DefaultEntity.38	Order04	12/1/2017 8:00:06 AM	12/1/2017 10:48:06 AM	2.8
5	Executing	DefaultEntity.37	Order03	12/1/2017 8:00:15 AM	12/1/2017 10:06:15 AM	2.1
7	Executing	DefaultEntity.36	Order01	12/1/2017 8:00:17 AM	12/1/2017 11:30:17 AM	3.5
8	Executing	DefaultEntity.39	Order05	12/1/2017 9:30:05 AM	12/1/2017 12:00:00 PM	2.49842
9	Executing	DefaultEntity.37	Order03	12/1/2017 10:06:15 AM	12/1/2017 10:18:15 AM	0.2
10	Executing	DefaultEntity.37	Order03	12/1/2017 10:18:22 AM	12/1/2017 10:30:22 AM	0.2
12	Executing	DefaultEntity.37	Order03	12/1/2017 10:30:22 AM	12/1/2017 12:00:00 PM	1.49366
13	Executing	DefaultEntity.38	Order04	12/1/2017 10:48:06 AM	12/1/2017 11:00:06 AM	0.2

Create Object From This – Enhancement

In Sprint 165, we added a right click option for objects that created a new sub-classed object with the object instance property values. We have now added a second right click option which will only be seen for objects that were originally created from the 'Create Object From This'. It appears if the object instance you opened it on has it's object/model in the current project.



Clicking it should update the property defaults on the object/model (and the defaults for any associated nodes), as well as update the property values any existing objects who hold values equal to the previous default to the new default. It WILL NOT update any graphical symbols, or “graphical” property options (for example, Current Symbol Index).

Simio Release 10 – Sprint 171 – May 1, 2018

This sprint, we have fixed several customer reported issues within the Flow Library as well as with breakpoints. Our development staff is currently working towards new features for supply chain modeling and continuing to enhance the Simio portal edition for cloud computing.

Simio Release 10 – Sprint 170 – April 10, 2018

This sprint, we have added several enhancements to the timer element for arrival tables, making them more flexible. Additionally, Enterprise users will find Gantt chart enhancements based on new task 'states'. All these sprint changes have been based on user requests.

Timer Element for Arrival Tables – Enhancements

We have added/changed several features within the *Interval Type* 'ArrivalTable' for the Timer element. These changes are also reflected within the Source object.

The Timer element now will exclude unrelated rows if an arrival table used to schedule timer events has primary key/foreign key relationships. There is a compatibility setting *Exclude Unrelated Rows In Timer Arrival Tables* is set to 'False' for all models built prior to sprint 170 and 'True' for all models built in 170+. This will allow a single arrival table to be referenced from multiple source objects using foreign key reference. For example, if the model has Store1, Store2 and Store3 sources, a single arrival table may include the arrival time of orders but include all the various stores.

The *Reset Event Name* property is now available if a Timer's *Interval Type* is 'ArrivalTable'. Resetting a Timer that is using an arrival table to schedule events simply resets its EventCount (may be pertinent if using the Maximum Events stopping condition) and resets its ElapsedTime state value.

If the *Repeat Arrival Pattern* property is 'True', the Timer now re-evaluates the *Arrival Events Per Time Slot* property at each discrete time in the table on every recurrence. Thus, you can now reasonably specify a random distribution for the number of arrival events per time slot. Previously, that property was evaluated once at initialization, and the number of arrival events for the time slot was then fixed to that initial value on each recurrence.

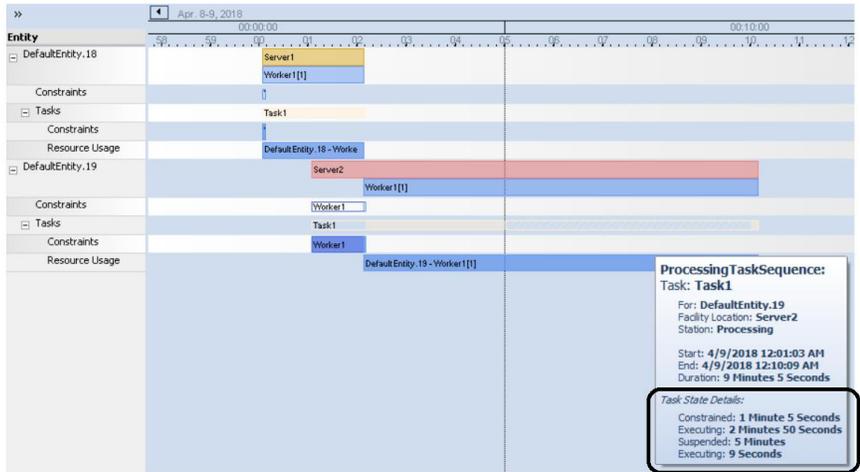
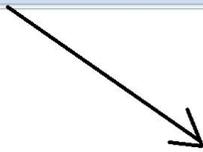
Task Log and Gantt Enhancements – Task State (Enterprise Edition)

We have added a Task State Log and incorporated the information to the Task Id field within the Task Log. There are now three possible task states in the log and associated Gantts:

- 1) *Executing* – The task was executing (incurring the specified processing time). - SOLID
- 2) *Constrained* – An active constraint was holding up execution of the task (e.g., waiting time for a secondary resource or some material). - EMPTY
- 3) *Suspended* – The execution of the task was suspended (this will happen if the primary resource or a secondary resource goes off-shift and the *Off-Shift Rule* is 'Suspend Processing'). - HATCHED

Below, you see the Task Log with the '+' for each task under Task Id column. Expanding that shows details of each task. Then, this translates into Entity Gantt, for example, showing empty area for *Constrained*, solid area for *Executing* and hatched area for *Suspended*. Note the second entity DefaultEntity.19. The task, Task1, shown is first *Constrained* waiting for the Worker1 to become available and move to the server. Then *Executing* of the process task begins. Before the processing task is complete, the Server2 goes off-shift and the task state becomes *Suspended*. Once back on shift, the remaining portion of the task is completed. The task state details are displayed in the box shown while hovering over the task.

Task Id	Entity Id	Entity	Facility Location	Station	Task Sequence	Task	Start Time	End Time	Duration (Hours)
1	DefaultEntity.18	DefaultEntity.18	Server1	Processing	ProcessingTaskSequence	Task1	4/9/2018 12:00:03 AM	4/9/2018 12:02:06 AM	0.0340944
Task State Details									
State									
	State	Start Time	End Time	Duration (Hours)					
	Constrained	4/9/2018 12:00:03 AM	4/9/2018 12:00:06 AM	0.000761111					
	Executing	4/9/2018 12:00:06 AM	4/9/2018 12:02:06 AM	0.0333333					
2									
	DefaultEntity.19	DefaultEntity.19	Server2	Processing	ProcessingTaskSequence	Task1	4/9/2018 12:01:03 AM	4/9/2018 12:10:09 AM	0.151454
Task State Details									
State									
	State	Start Time	End Time	Duration (Hours)					
	Constrained	4/9/2018 12:01:03 AM	4/9/2018 12:02:09 AM	0.0181211					
	Executing	4/9/2018 12:02:09 AM	4/9/2018 12:05:00 AM	0.0474564					
	Suspended	4/9/2018 12:05:00 AM	4/9/2018 12:10:00 AM	0.0833333					
	Executing	4/9/2018 12:10:00 AM	4/9/2018 12:10:09 AM	0.00254361					



Simio Release 10 – Sprint 169 – March 19, 2018

This sprint, we have added several user-requested enhancements, such as a new SimBit for demonstrating worker efficiencies with groups of workers, as well as further categorization for Entity Workflow Gantt charts. Our development staff is also working on many enhancements with our Simio Portal edition as well.

Show Commonly Used Properties Only feature – Removed/Obsolete

Based on customer feedback, we have removed the *Show Commonly Used Properties Only* checkbox that used to reside at the top of all objects. This feature, if checked (default value) would cause only a few of the properties of an object to be visible. This was originally added to enable beginner users to see smaller amounts of information when they first started using Simio objects. The associated *Commonly Used Property* for object properties has also been removed.



Resource Type Objects - Log Resource Usage Property Enhancement

Within any resource type of object, such as Server, Worker, Vehicle, etc, there is a *Log Resource Usage* property under the Advanced Options section of properties. This Boolean (True/False) property can now be mapped to a model property. This is useful if you want to, for example, set up different groupings of resources that can be selectively included in the Resource Gantt by toggling a single property on the model.

New SimBit

We have added a new SimBit that includes two separate models to demonstrate the use of worker groups and their efficiencies and skills. The project is named WorkerPoolWithEfficiency and includes the below models:

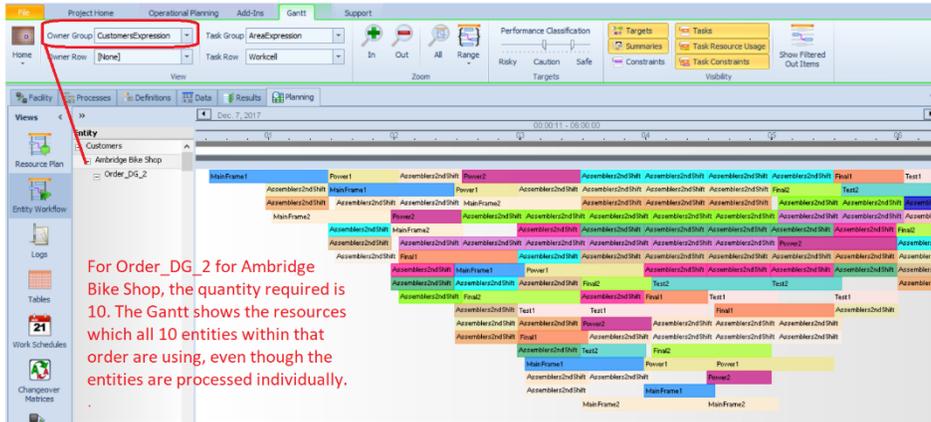
SimpleWorkerPool – Demonstrates how to incorporate individual worker walking speed and efficiency based on worker selection.

ComplexWorkerPool – Demonstrates how to incorporate worker skill level or ability to work at particular servers. The selection of the worker from a pool (list) is based on efficiency factor.

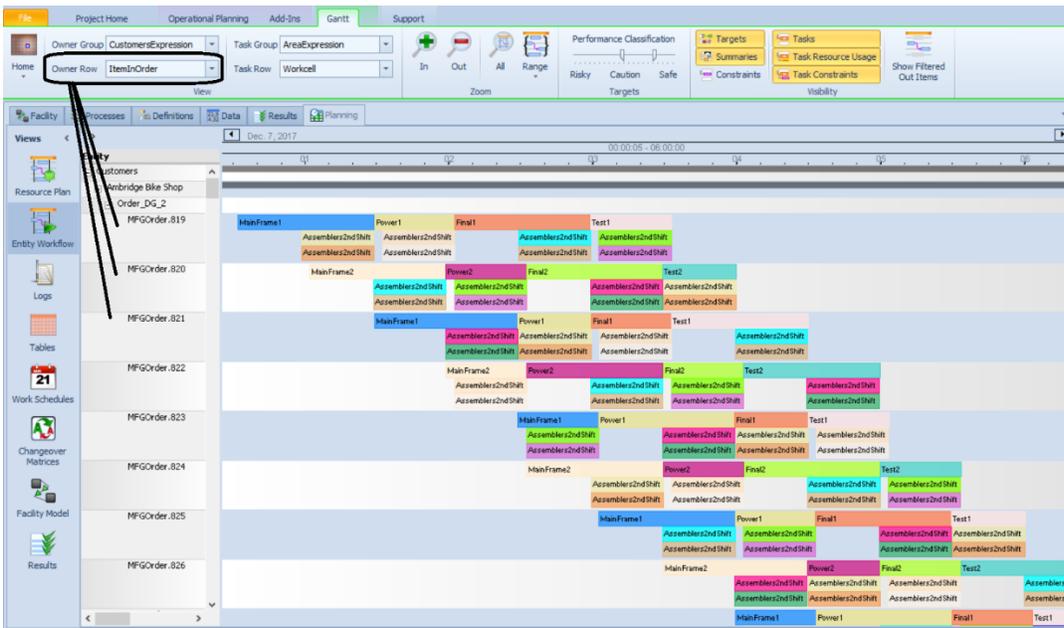
Entity Workflow Gantt Enhancement – Enterprise Edition

In September within sprint 162, we made some changes to the Entity Workflow Gantt to allow for task categorization using Task Group and Task Row within the Gantt ribbon. With this sprint, we've added a different type of categorization within the main entity portion of the Gantt to allow for additional categorization as well. The new *Owner Row* option within the Gantt ribbon allows the user to split out the "main" time items into sub-rows under the main row. For example, in the below SchedulingBicycleAssembly example, the default entity Gantt for a given 'order', which may consist of a quantity of entity items.

Order Id	Customer	Material Name	Release Date	Due Date	Order Status	Priority	Quantity
1	Order_SG_1	Walmart	EcobikesTDGreen	12/4/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	12
2	Order_SW_1	Walmart	EcobikesTDWhite	12/4/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	10
3	Order_DG_1	Walmart	EcobikesTDGreen	12/8/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	8
4	Order_DW_1	Walmart	EcobikesTDWhite	12/4/2017 8:00:00 AM	12/8/2017 4:00:00 PM	New	9
5	Order_SG_2	Ambridge Bike Shop	EcobikesTDGreen	12/4/2017 8:00:00 AM	12/7/2017 8:00:00 AM	New	15
6	Order_SW_2	Ambridge Bike Shop	EcobikesTDWhite	12/4/2017 8:00:00 AM	12/7/2017 4:00:00 PM	New	9
7	Order_DG_2	Ambridge Bike Shop	EcobikesTDGreen	12/9/2017 8:00:00 AM	12/7/2017 7:00:00 PM	New	10



Users can now specify a Resource Usage Log custom column for Owner Row that will generate sub-rows based on the column information. For this example, the item (or entity) within the order is used as the Owner Row.



This feature is especially useful for entities that have multiple components of processing occurring at the same time with multiple workers/resources also allocated at that location – the location itself can then be used as the Owner Row.

Simio Release 10 – Sprint 168 – February 23, 2018

This sprint, we have relaxed the step limitations on Decision type processes, allowing user steps, such as Write step, to be used. We've also updated some graphics settings within the Simio Application Settings area.

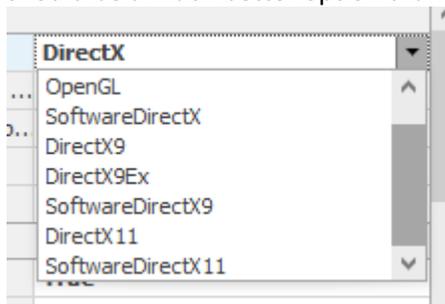
Decision/OnRunEnding Processes - Remove Step Type Restrictions with Continued Error If Any Token Delay

Previously, Simio allowed only a specific set of steps (for example Assign, Decide) to be used in a decision process (e.g., the OnEvaluatingSeizeRequest process of a resource) or an OnRunEnding process of an object. The step type restrictions have been removed (thus allowing steps, such as user-defined Write step, per customer request). Simio continues, however, to issue an error when a token is delayed for any reason, such as with a Delay or Wait step, within a decision or OnRunEnding process.

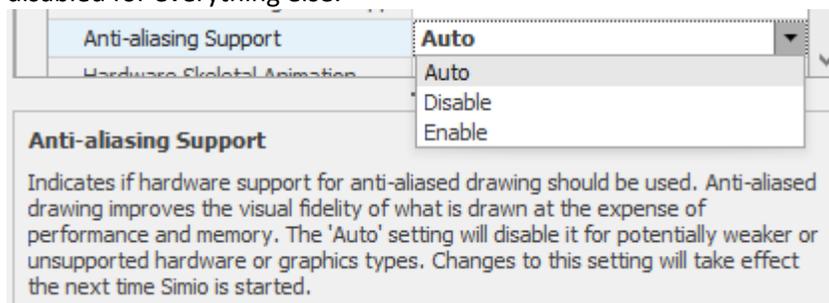
Simio File > Settings ... Graphics - DirectX 11, Anti-aliasing, and Hardware Skeletal Animation Support

There are now a few new options in the File -> Settings dialog related to graphics:

1. An expanded set of options for the *Graphics Type*. Note that "DirectX" or "SoftwareDirectX" (currently) actually mean DirectX 9. Users also can explicitly say which DirectX they want. Note also that "SoftwareDirectX11" is using the DirectX WARP driver ([https://msdn.microsoft.com/en-us/library/windows/desktop/gg615082\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/gg615082(v=vs.85).aspx)), which should be a much better option than the existing SoftwareDirectX9 for VM.



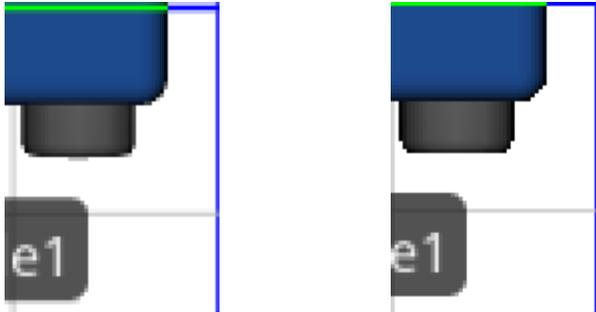
2. A new *Anti-aliasing Support* option. When 'Auto' is selected, will be enabled for DirectX 11 and disabled for everything else.



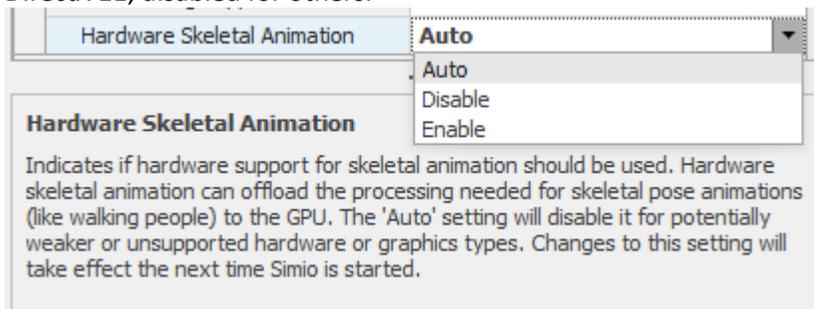
Some zoomed in screenshots:

Anti-alias on:

Anti-alias off:



3. A new *Hardware Skeletal Animation* option. When in 'Auto' mode, will be enabled by default for DirectX 11, disabled for others.



Simio Release 10 – Sprint 167 – February 2, 2018

With Sprint 167, we continue to enhance many of our Standard Library objects and their internal steps/elements with user requested features. This includes multiple enhancements to the BatchLogic element and related Combiner object, as well as new options on the RoutingGroup element and Suspend/Resume steps.

RoutingGroup Element – On Confirming Destination Node Assignment Process

Within the Advanced Options of the RoutingGroup element, we have added a new process that may be called, *On Confirming Destination Node Assignment Process*. This was a user requested feature.

When using a dynamic selection rule, the existing *On Evaluating Route Request Process* is executed for every possible entity and destination node combination, to filter the list of possible destination node assignments. The best destination node assignment from the candidates that made it through the filtering is then selected per the specified rule.

The new *On Confirming Destination Node Assignment Process* on the other hand is only executed whenever a destination node assignment has been selected, to confirm it. If this decision process returns False, then the routing group goes back to the drawing board and using the dynamic selection rule again looking for the next best entity/node combination (which would then have to be confirmed and so forth).

The screenshot shows the 'Properties: RoutingGroup1 (Routing Group Element)' window. The 'Advanced Options' section is expanded, showing the following properties:

Destination Node List Name	
Route Request Ranking Rule	FirstInFirstOut
Route Request Dynamic Selection Rule	None
Evaluate Route Requests Triggers	0 Rows
Destination Blocked Condition	Candidate.Node.Assigned
On Evaluating Route Request Process	
On Confirming Destination Node Assignment Process	
On Destination Node Assigned Process	

Three callout boxes are present:

- On Evaluating Route Request Process**: Optional decision process to be executed by the routing group whenever it is evaluating whether an entity is eligible to be assigned a destination node. In the executed decision process, assigning a non-positive value to the executing token's ReturnValue state indicates that the route request is not eligible.
- On Confirming Destination Node Assignment Process**: Optional decision process to be executed by the routing group to confirm a selected destination node assignment for an entity. In the executed decision process, assigning a non-positive value to the executing token's ReturnValue state indicates to cancel the destination node assignment and look for a different selection (either entity or node). **NEW**
- On Destination Node Assigned Process**: Optional process to be executed whenever an entity has been assigned a destination node by the routing group.

BatchLogic Element/Standard Library Combiner – Multiple Enhancements

The BatchLogic element and related Combiner object within the Standard Library have several enhancements based on multiple user requests. These include:

1) Allow Parent & Member Match Expressions To Return Values Of Any Data Type (e.g. Strings)

Prior to this sprint, users could only batch entities by matching numeric attributes. This enhancement will allow matching attributes of any data type (e.g., strings). Multiple users have requested this feature.

For example, you can now specify the *Matching Rule* for a Combiner or BatchLogic element as 'Match Members' and specify the *Member Match Expression* as 'Entity.EntityType', which means to only group members together that are of the same entity type reference. Or, as another example, specify the *Member Match Expression* as 'Orders.OrderID' where that table value to group member entities together is an order identifier string.

Note that the trace detail for the Batch step/BatchLogic element was also to provide more detail in the trace of the match expression values being used to do the batching.

2) Handling Simultaneous Entity Arrivals - Schedule Late Current Event To Try Batching When Batch Queue Item Inserted

In Run Setup -> Advanced Compatibility Settings, a *Schedule Late Current Event To Try Batching When Batch Queue Item Inserted* compatibility setting was added.

This setting indicates whether to schedule a late priority current event to try batching whenever a new item has been inserted into the parent or member queue of a batch logic element. Otherwise, batching will be immediately attempted before the execution of any other simulation logic in the system. To better handle simultaneous entity arrivals to batching queues, the recommended value for this setting is 'True'.

This setting is 'False' by default for models saved in Sprint 166 version or older, which was the implicit setting up to that point. The setting is 'True' for new models.

The information above was added to the Simio Compatibility Notes document.

Suspend / Resume Steps – Enhancement

Both the Suspend and Resume steps now allow the ability to suspend and resume batching done within a Batch Logic element. The *Suspend Type* and *Resume Type* properties on these steps now include the option for 'BatchLogic'.



Standard Library Combiner – New Suspend Batching When Down Property

There is a new property for the Combiner object within the Standard Library, shown below within the Batching Logic section of properties. When the *Suspend Batching When Down* is set to 'True', the actual collection of batches will be stopped if the Combiner has failed or changes capacity to zero from a work schedule shift change. This means that the parent and member entities will reside in the input queues of the Combiner and not attempt batching. If this new property is set to 'False', parent and member entities will continue the 'batching' process and will be removed from the parent and member input queues but will not move through the Combiner object until the failure is repaired or work schedule changes.

*Note: The default value of this property is 'True' for new models, but 'False' if updating an old model (as that was the implicit setting in the previous object versions).



Model Properties/Compatibility – Parse Non-Entity Associated Objects for Property Reference

Within a model's property window, under the Advanced Options / Compatibility section of properties, we have added a Parse Non-Entity Associated Objects for Property Reference property. In models built in sprint 166 and prior, this value will implicitly be set to 'False'. This was a user requested issue that changed previous software behavior.

Now within Sprint 167+, an Object, such as a custom server named MyServer for example, may have a property, such as a Table type property named *MyTable*, that can be referenced within the main model when that object is placed. Thus, in the above example, a SetRow step may reference 'MyServer.MyTable' as the *Table Name* property to set a particular table (based on the associated object MyServer's property value for *MyTable*) row reference for any of the MyServer objects placed in a model.

Simio Release 10 – Sprint 166 – January 11, 2018

With sprint 166, we have enhanced the capability of several of the steps with multiple token exits and have made several enhancements to the Search step. We have also added support for balking and renegeing from a member output queue on the Separator object. Several additional user requested features have also been added.

Separator – Buffer Logic support for Balking / Reneging from Member Output Buffer

Within sprint 146, we added support for balking and renegeing from most input/output buffers within the Standard Library objects. However, one buffer that this did not include was the member output buffer for the Separator object. This object has now been enhanced to include the balking/renegeing features as well.

New Token Wait Action for Create, Remove, UnBatch, and Search steps

The Create, Remove, UnBatch and Search steps all consist of the original (or parent) token exit from the right of the step, as well as the created/removed/member/found exit from the bottom of the step. By default, the created/removed/member/found tokens will exit first and continue until a delay in processing, followed by the original/parent token processing. The use of a Delay step with Math.Epsilon could have been used to change this processing order.

These steps now include a Token Wait Action property to help users determine how the tokens exit the step. For compatibility, the default value of this property is 'NewTokensExitFirst' which allows the tokens to exit/process as they always have. Two additional options are 'NoWait' and 'WaitUntilNewTokenProcessingCompleted'. 'NoWait' indicates that the execution of any new tokens (from created/removed/member/found exit) will be scheduled on the simulation's current event calendar as an early priority event and that the original (or parent) token will immediately exit the step first. If the action is 'WaitUntilNewTokenProcessingCompleted', the original token will wait until the processing has been completed for all new tokens associated with created/removed/member/found before exiting. This includes any Delay steps of Math.Epsilon that the new tokens may encounter.

These new options were added at the request of multiple customers at our user's meeting this past May.

Search Step Enhancements

In addition to the above mentioned 'Token Wait Action' feature, we have made several other changes to the Search step functionality, including

- 1) The *Collection Type* 'ReservationOwners' is now available to search the list of objects that have currently reserved capacity of a specified resource.
- 2) If searching table rows, the table row references of the original token are now copied to any new tokens. Search steps in models built prior to Sprint 166 will run as they had and not include this functionality.
- 3) The *Search Types* 'Forward' & 'Backward' no longer sum a specified *Search Expression*. Those search types are now simply for the most common use case of searching a collection of objects or rows of a table with an optional *Match Condition*. This simplifies the default presentation of the Search step's properties by no longer showing *Search Expression*.

New 'ForwardSumExpression' & 'BackwardSumExpression' search types are now provided if

the expression summing behavior is desired. Search steps added prior to Sprint 166 will now have 'ForwardSumExpression' instead of 'Forward' and 'BackwardSumExpression' instead of 'Backward' for compatibility reasons.

- 4) In conjunction with the above change, a new *Save Search Expression Sum* property is now provided along with the *Search Expression* property if the search type is 'MinimizeExpression', 'MaximizeExpression', 'ForwardSumExpression', or 'BackwardSumExpression'. This new approach was done to make it easier for users to both understand step behavior as well as to avoid unit conversion mistakes because we can save the search expression sum in a state variable of some unit type.
- 5) The *Limit* property has been moved from the Advanced Options section of properties to the Basic Logic section of properties, as requested by several users.
- 6) While making the other enhancements in this list, a small performance improvement in the code was identified if searching table rows. Simple test models run about 5-6% faster after the change.

State Statistic Element Enhancement

This element now allows for non-static *Display Category* value for use for the State Statistic element in the Gantt. This provides users the capability of using table data to categorize auto-created state statistic elements.

RidePickupQueue and Other Queue Statistics Changes

Queue statistics were the only type of statistics in that engine whereby, if the *Report Statistics* property of the parent object or element was set to 'False', then those statistics were neither collected or reported to the Results. No other statistics behaved that way, such as Tally Statistic or resource or entity population statistics. That behavior was sometimes causing user confusion.

Now, ALL statistics in the product are always collected. Thus any statistic function being referenced in an expression (such as animating `SomeTransferNode.RidePickupQueue.AverageTimeWaiting`) will always return the expected value.

The *Report Statistics* property behavior is now consistent across the board. For any object or element, that property simply filters out any automatic statistics reported by that object or element to the Results view. By default, the *Report Statistic* property for `BasicNode` and `TransferNode` is set to 'False', to simplify the default Results view presentation.

The potential confusion with using the queue statistics functions has now been eliminated.

New SimBit

We have added a new SimBit to the selection of small example models provided with the software.

DisplayingEntityDynamicStatusInformation – This SimBit displays multiple label types (dynamic label text as well as status label) that can be attached to an entity as it moves through the system. The example shows multiple lines of data for the entities and uses both strings and string functions to display the information.

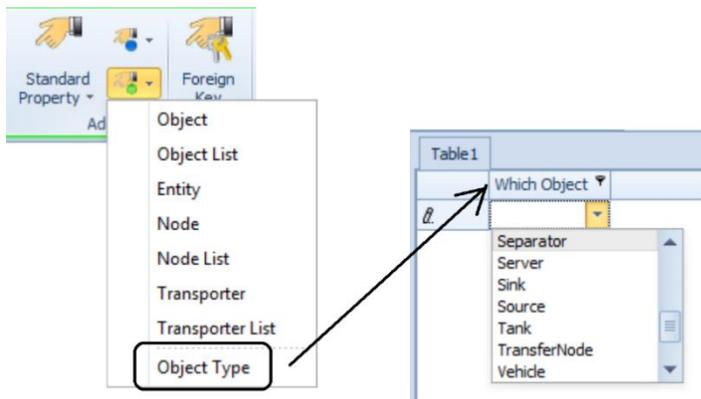
Simio Release 10 – Sprint 164/165 – November 20, 2017

In the last sprint, we added the auto-create capability to allow objects within the Facility window to be created by table data. This sprint, we have added many features to expand on that functionality. For example, we have added the object reference type of Object Type so that different types of objects can be automatically created from a single column in a table.

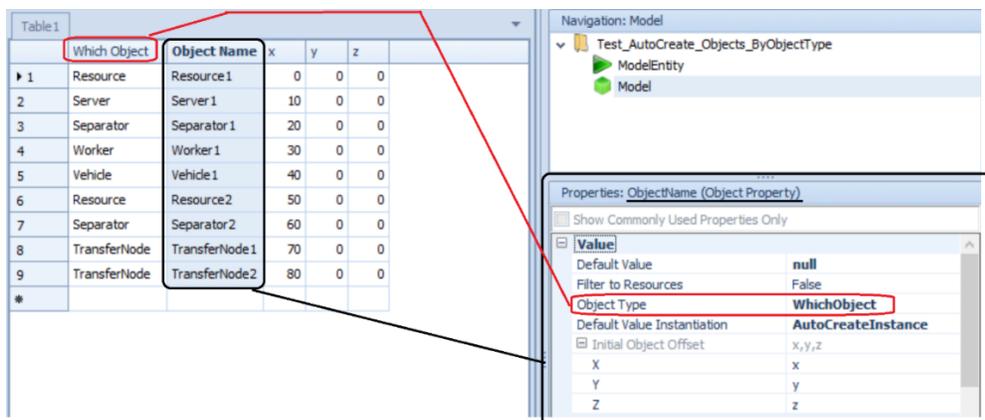
We've also added the ability to right click on an object and change the type of object – let's say from a Resource to a Worker. Within repeating group properties, users can now define 'default tuples' which are default values for one+ repeating property tuple. All these features, along with a right-click option to 'Create Object From This' object instance to subclass an object with its default data, provide great flexibility for data driven modeling.

New 'Object Type' Column for Data Tables

We have added Object Type to the list of object reference types for columns within a table. This then allows table column values to reference objects from the project and any loaded libraries.



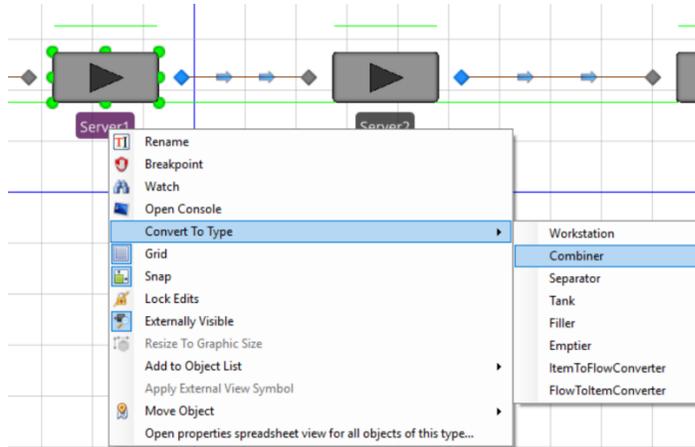
Then, a column with object names that is set to 'auto-create' objects within the Facility window (such as 'ObjectName' column below) can reference the Object Type column (such as 'WhichObject' column below) for the type of object to create.



'Convert To Type' – Right Click Option

We have a right-click option within the Facility window to Convert To Type > *** based on the type of current object that is selected. This will allow users to easily convert currently placed and connected objects to other type of objects. For example, a user may decide to change a Resource object to a Worker type object or to convert a Server to a Combiner.

Note that conversions are context specific to the number of input/output nodes with links, so an unconnected Server can be converted to many different objects, while a connected Server will only provide types with at least as many nodes as needed to keep the attached links (see below).



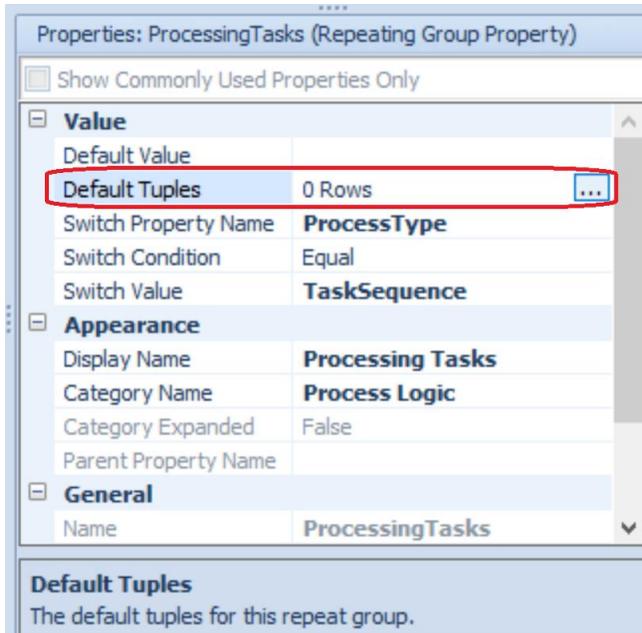
Note that the property value copying works as it currently does, which is to go to the first common shared base class of the two types, and copy those property values from the old instance to the new instance. So, converting between Server -> Combiner will convert all properties of Fixed, but converting Server -> Vehicle will convert all properties of Object.

Note that we also keep our current naming behavior, which is that if the old instance name looks like an auto-name (like 'Server1'), we keep the auto-name of the new instance (like 'Combiner1'), but if it ***isn't*** an auto-name (like 'PaintingArea'), we keep that same name for the new instance.

Repeat Group Tuples – New Default Values

We now allow for specifying default values for repeating group properties within repeat groups. We have added a new *Default Tuples* property to the repeat group definition which will open a tuple/repeating property editor, where each tuple will be a list of property definitions under the repeat group. Just like in the regular *Default Value* property, the user only enters strings and there is no editing context or expression validation.

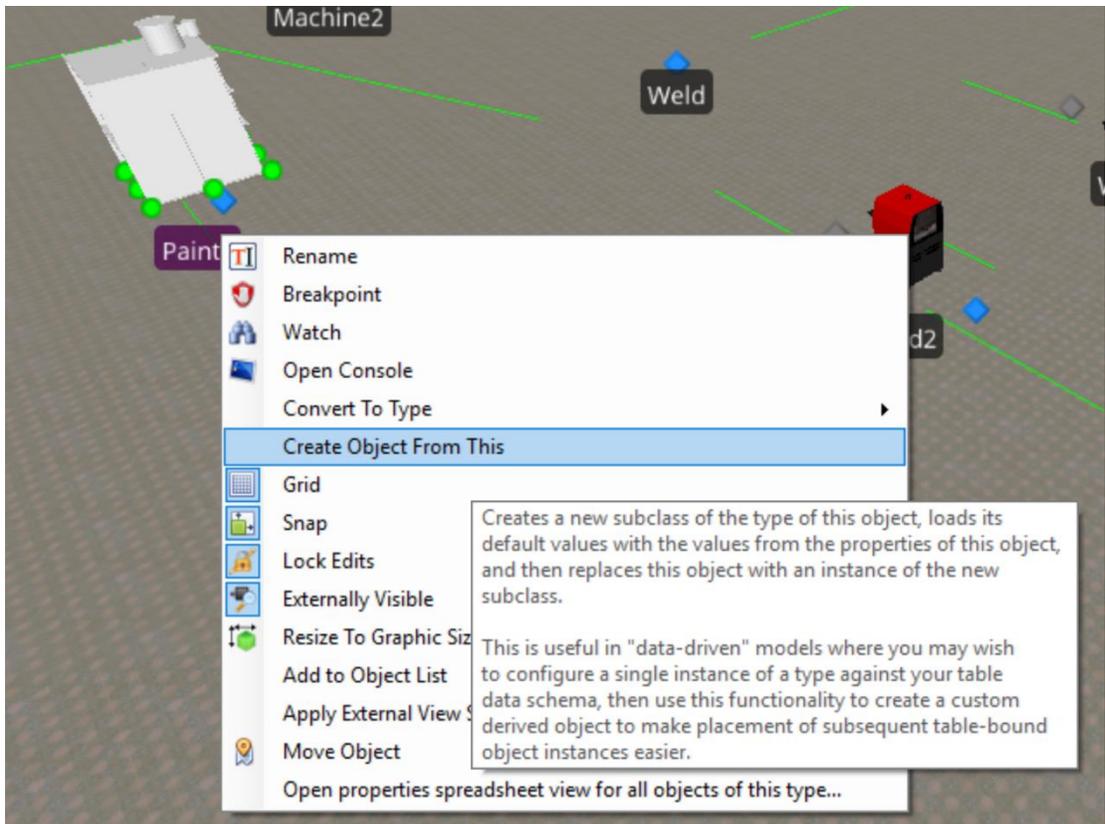
In the example below, a Server was 'sub-classed' and within the properties of the new sub-classed MyServer, the ProcessingTasks property for the Server is a repeating group property. The new *Default Tuples* property can be seen circled in red below. Each *Default Tuples* property, when opened, will display the properties that exist within the repeat group. These properties may then have default values that either 1) differ from the object default value and/or 2) reference table values or other expressions. N number of tuples can be populated. This functionality can then be used to 'pre-populate' data, such as task information, into objects.



'Create Object From This' – Right Click Option

We now have a right click option on objects called 'Create Object From This'. This feature will allow users to enter property information into a Server, for example, including processing tasks, secondary resources, material constraints, capacity/workschedule information, etc. and then right click to create a 'subclassed' object of the Server (MyServer) that includes all the property values as 'default' information. It will also then change that object to the new subclass object type (MyServer).

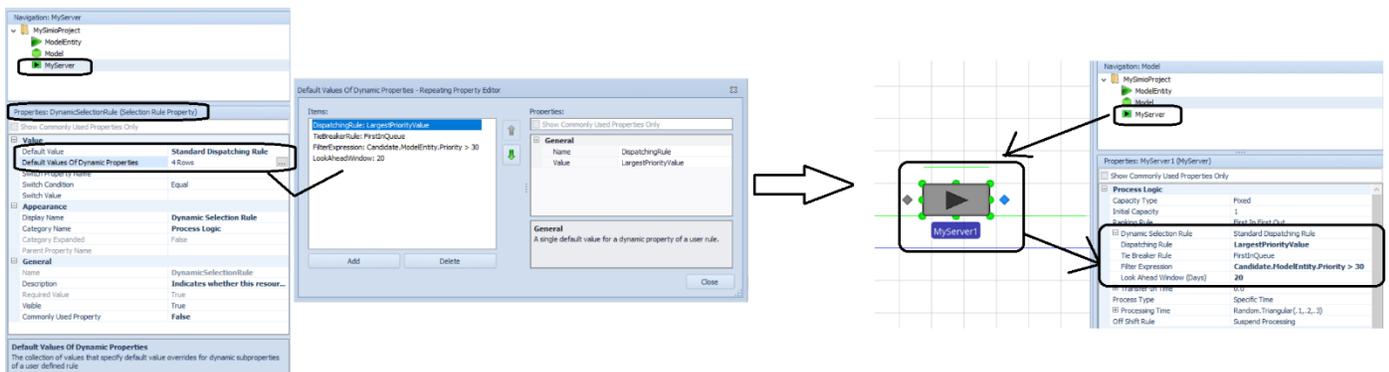
Note that we also have API support for this feature as well.



Dynamic Sub Properties (Various Rule Properties) – New Default Values

Along the same lines as the above feature, this enhancement adds new *Default Values of Dynamic Properties* repeating group property right under the *Default Value* property in the property grid for Selection Rule and Travel Steering Rule type properties.

So, when an object property, such as Dynamic Selection Rule, has a *Default Value* of 'Standard Dispatching Rule', several additional properties, such as *Dispatching Rule* and *Tie Breaker Rule* become available for the user. In a typical Server object, the default values for these are 'FirstInQueue'. To change those default values for an object, such as a sub-classed MyServer (below), the user would add the property *Name* and new default *Value* in the repeating property editor.



New Station Function

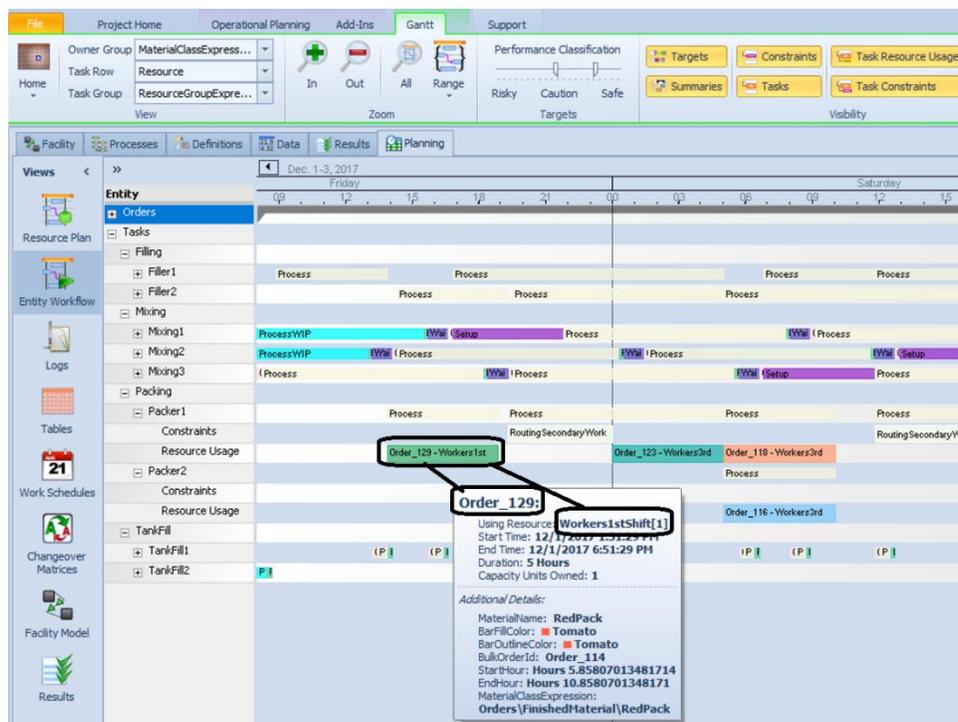
NumberTransferringOut - Returns the current number of entities attempting to transfer out of the station.

Release step & Standard Library Resource Releases Enhancement

We have enhanced the *Keep Reserved If* and *Reservation Timeout* properties to allow Candidate.[ResourceClass].[Attribute] syntax in the expression. This enables these property values to be based on an attribute of the resource(s) being released, such as resource capacity or the number of entities in a resource allocation queue.

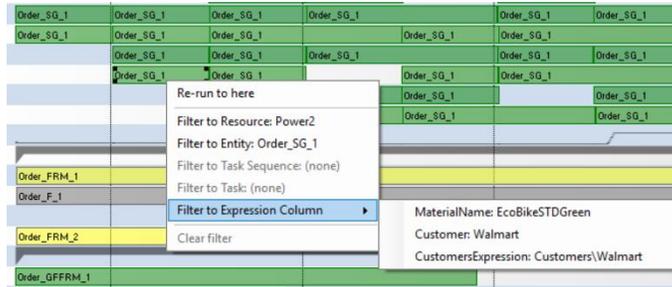
Enterprise Gantt Resource Usage Bars - Concatenate Owner and Resource ID from Resource Usage Log

We have enhanced the Gantt charts so that when the Tasks are expanded, the Resource Usage sections of the Gantt bars show both the Entity information (owner of the resource) as well as the Resource name.



Enterprise Gantt Filtering

We have also added the ability to right click on an area within either the Resource Plan or Entity Workflow Gantt and filter to an expression column that was added to the Resource Usage Log. These expressions are shown in the Additional Details section when hovered. Now, they can be used to filter the Gantt as well.



Local Language Support

We have added Russian to the list of languages supported by the Simio software. Simio now supports Bulgarian, Chinese, Czech, English, French, German, Japanese, Korean, Portuguese, Russian, Spanish, and Turkish using the *-language:XX* command line switch. Some of these have been manually translated by our users and others rely on automatic machine translation. The automatic parts of the existing translations have been updated using the latest Microsoft translation technology.

Simio Release 9 – Sprint 163 – October 20, 2017

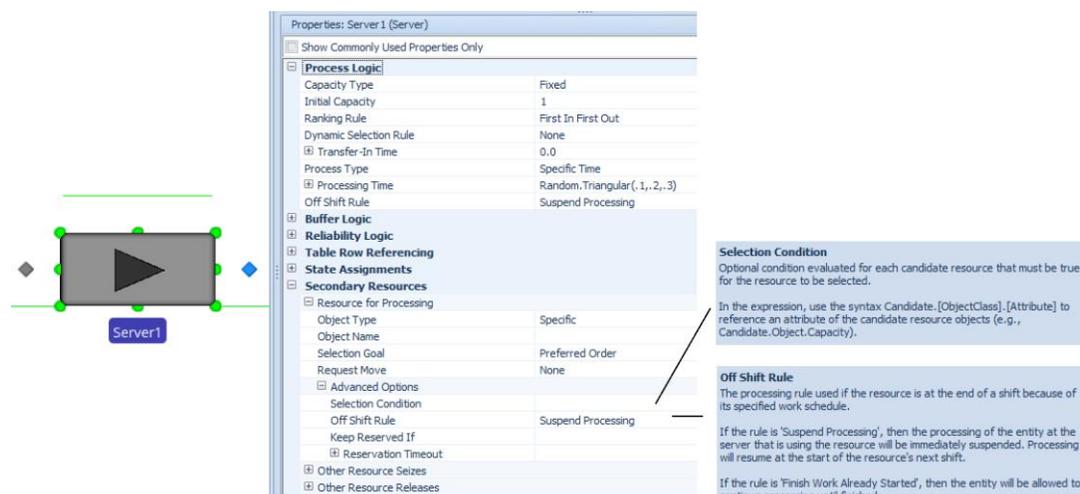
In this sprint, we have many user-requested enhancements to Simio. First, a new *Off Shift Rule* is now available for secondary resources, as well as task-related resources. Resources that go off shift can either suspend processing or finish the work already started. Additionally, many resource reservation functions have been added, in addition to a *Task Ready* add on process for task processing. Interactive Gantt charts are now available within the Enterprise edition in the Results tab area. And finally, the auto-create capability has been added to table columns for object references to allow objects within the Facility window to be created by table data. Look for more enhancements for this capability in the next sprint as well! Exciting stuff!

New *Off Shift Rule* for Secondary Resources

We have enhanced the Server, Combiner, Separator, Filler and Emptier objects in the Standard and Flow Libraries to include an *Off Shift Rule* property for the secondary resources used for processing within the object.

If an entity's processing requires the Secondary Resources -> Resource for Processing or if an entity's processing is a task sequence, and resource requirements are specified for any task, the *Off Shift Rule* property allows a user to easily indicate whether to 'Suspend Processing' or 'Finish Work Already Started' if the seized resource is at the end of a shift because of a specified work schedule.

The below diagram shows the *Off Shift Rule* used for Secondary Resources for processing.

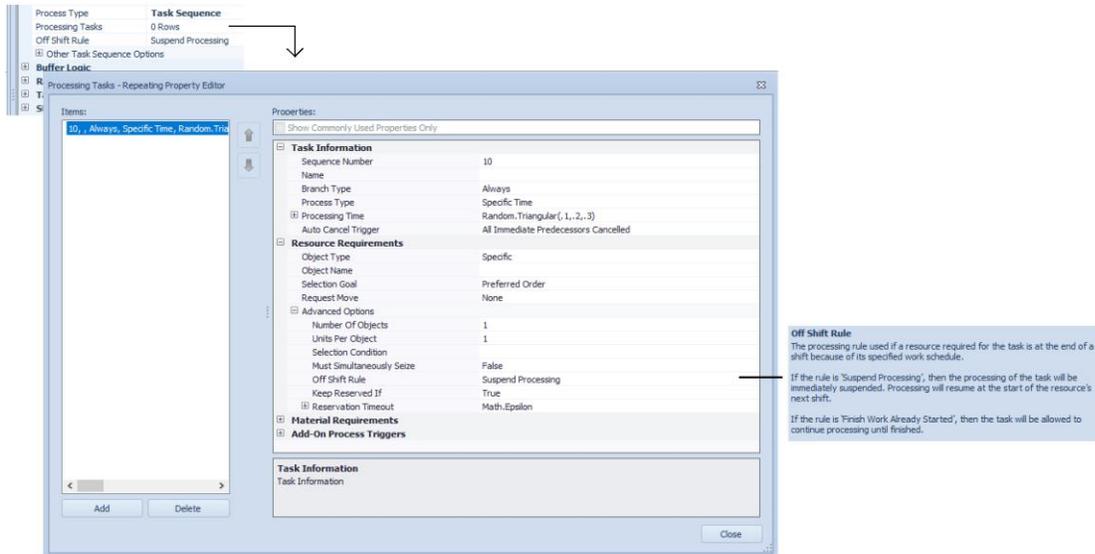


The screenshot shows the Properties window for 'Server 1 (Server)'. The 'Secondary Resources' section is expanded, showing the 'Resource for Processing' table with 'Object Type' set to 'Specific' and 'Off Shift Rule' set to 'Suspend Processing'. A callout box on the right explains the 'Off Shift Rule' and 'Selection Condition'.

Selection Condition
Optional condition evaluated for each candidate resource that must be true for the resource to be selected.
In the expression, use the syntax Candidate.[ObjectClass].[Attribute] to reference an attribute of the candidate resource objects (e.g., Candidate.Object.Capacity).

Off Shift Rule
The processing rule used if the resource is at the end of a shift because of its specified work schedule.
If the rule is 'Suspend Processing', then the processing of the entity at the server that is using the resource will be immediately suspended. Processing will resume at the start of the resource's next shift.
If the rule is 'Finish Work Already Started', then the entity will be allowed to continue processing until finished.

The below diagram shows the *Off Shift Rule* for resources used for processing tasks.



Resources, Workers and/or Vehicles can be used as secondary resources. If seized as a secondary resource for one or more processing task(s) at a Server, Combiner, Separator, Filler, or Emptier, and at the end of a shift because of a specified work schedule, the resource's ResourceState will be assigned to 'OffShift' if all related processing is suspended. Otherwise, the ResourceState will be assigned to 'OffShiftBusy'.

New Resource Functions – Reserved Resources

ReservationOwners - Provides functions for accessing the objects that currently have reservations for capacity units of this resource.

ReservationOwners.NumberItems - Returns the number of objects that currently have reservations for capacity units of this resource.

ReservationOwners.FirstItem - Returns a reference to the first object in the list of objects that currently have reservations for capacity units of this resource.

ReservationOwners.LastItem - Returns a reference to the last object in the list of objects that currently have reservations for capacity units of this resource.

ReservationOwners.ItemAtIndex(index) - Returns a reference to the object at a specified index position in the list of objects that currently have reservations for capacity units of this resource.

ReservationOwners.IndexOfItem(owner) - Returns the one-based index of the first occurrence of a specified object in the list of objects that currently have reservations for capacity units of this resource. If the object has not reserved, the resource then the value 0 is returned.

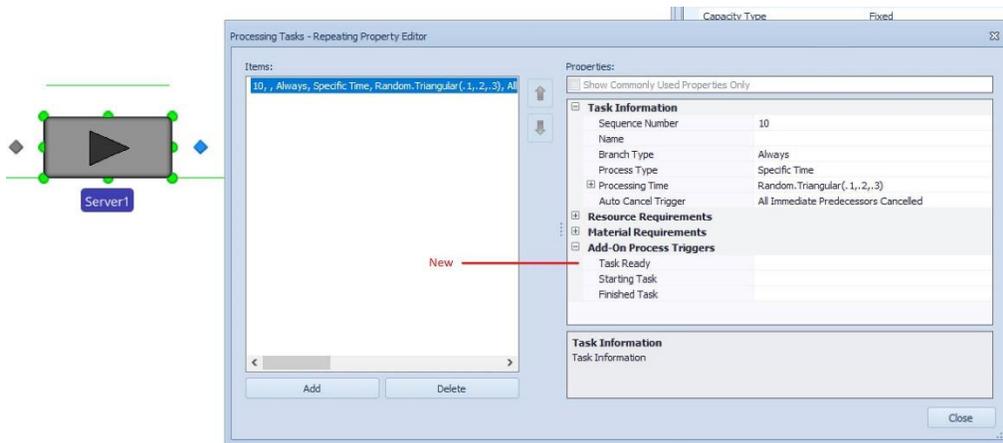
ReservationOwners.Contains (owner) - Returns True (1) if the objects that currently have reservations for capacity units of this resource include the specified object. Otherwise, the value False (0) is returned.

New Object Function – SeizedResources

SeizedResources.RequestedDestinationNodeFor(resource) - Returns a reference to the requested destination node for a specified resource in the list of resources currently seized by this object. If the resource is not found or was not requested to move when seized then the Nothing keyword is returned.

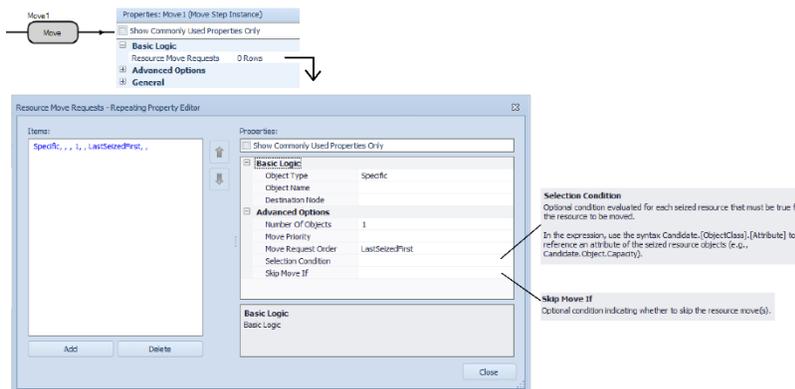
Task Sequences - New *Task Ready* Add-On Process Trigger

We have added a new *Task Ready* add-on process trigger to the Processing Tasks repeating property editor for Task Sequences. This was added per customer request to provide an add-on process before the attempt to seize or consume a task's resource and material requirements. This process will be executed when all the task's predecessor dependencies have been satisfied.



Move Step - New *Skip Move If* Property

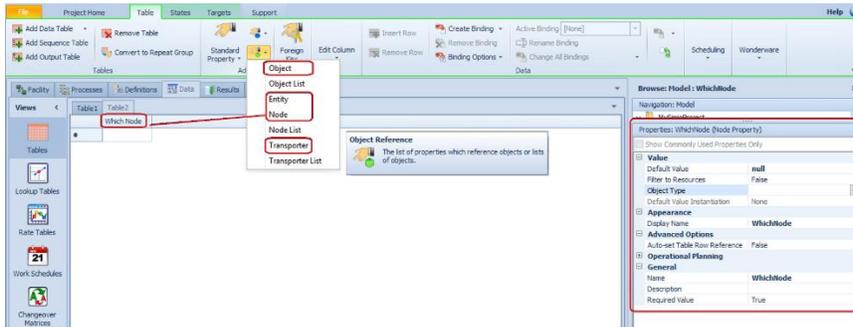
We have added a *Skip Move If* property for the Move step which is an optional condition that will be evaluated for each resource that has a move request. If the expression evaluates to false, the move will not take place. Also within the Move step, we have changed the name of the property *Move Request Condition* to be *Selection Condition* for consistency among steps.



Data Tables Enhancement – Auto-Create for Object Reference Columns

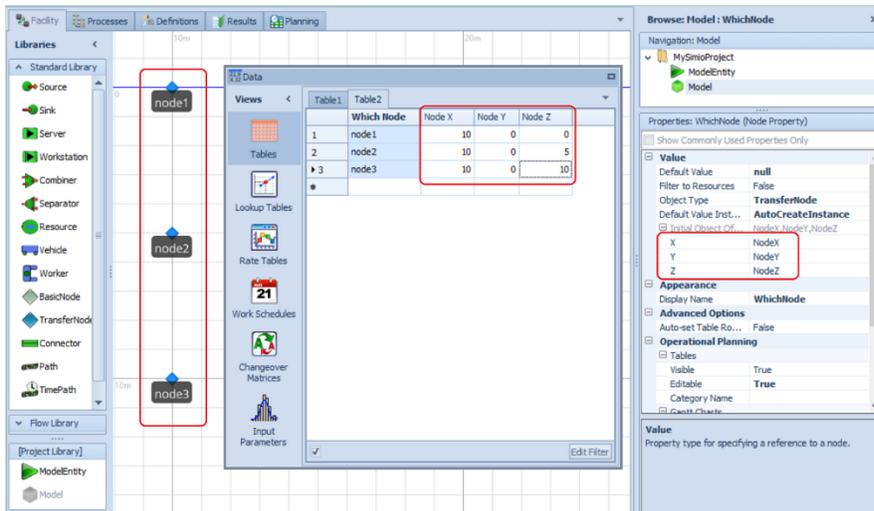
We have added the capability to auto-create objects by using the Object Reference column type within a data table or sequence table. When a column is of type Object Reference (i.e., Object, Entity, Node or Transporter), there are additional properties, including *Object Type*, *Default Value Instantiation* and

Initial Object Offset values that can be used to automatically create specific objects within the Facility window.



When the *Object Type* property of the Object reference column is specified as an object, such as 'TransferNode' for a Node reference or 'Worker' for a Transporter reference, then the *Default Value Instantiation* property becomes editable. When *Default Value Instantiation* is set to 'AutoCreateInstance' (instead of 'None' which is the default value), then all objects specified within the Object reference column of the table will be automatically created.

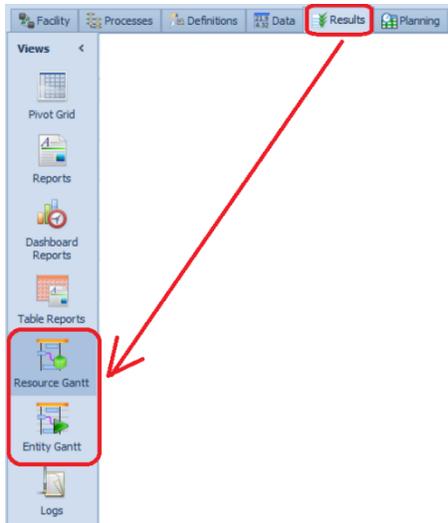
The *Initial Object Offset* values for X, Y, Z locations in the Facility window can be specified as the default of 0,0,0 or a real/integer value. These location values can also reference a real/integer column within the same table, where the objects have different location values and are placed accordingly within the Facility window, as shown below.



*Note: This is the first in several enhancements for easier data driven model building. *

Gantt Views in Interactive Mode – Enterprise Edition

We have enhanced the Results tab views to include the Resource Gantt and Entity Gantt charts. This will enable users to view the Gantts in interactive mode, including all randomness and failures incorporated in the interactive model. These Gantt charts are then based on the logs generated for the interactive run. *Remember that within the Gantt charts in the Planning tab, all randomness and failures are disabled.



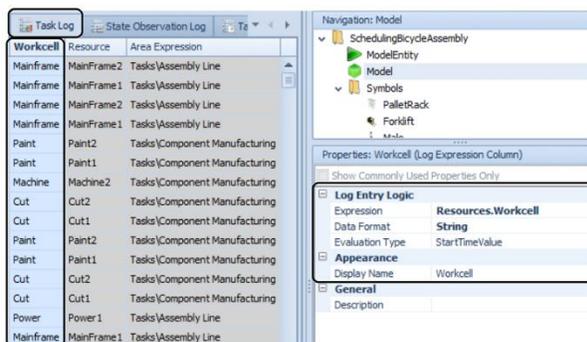
Simio Release 9 – Sprint 162 – September 21, 2017

In this sprint, we have added much functionality for categorizing and displaying tasks within the Gantt charts within Simio Enterprise edition. Additionally, many new functions have been added to both processes and tokens in preparation allowing secondary/task-related resources to have an *Off-Shift Rule*, which will be available in the next sprint.

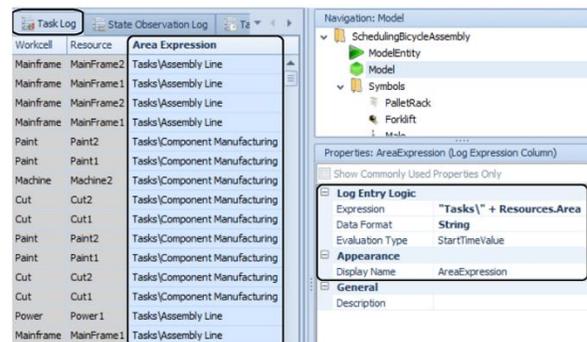
Simio Enterprise Edition – Task

When Processing Tasks type processing is used within many of the Standard Library objects, the user has the ability to specify multiple tasks, as well as required resources and/or materials within the Processing Tasks repeating property for an object (Server, Combiner, Separator, for example). Information about the task starting/ending times, as well as other related information is stored in the Task Log. In the previous sprint, we added a unique Task Id for each task which allows then Resource Usage, Material Usage, Transporter Usage and Constraint logs to have associated 'tasks'.

Within each Task Log (as with the Resource Usage Log), additional columns may be added to the log to provide additional information for Gantt charts. Below, the Workcell column was added to a Task Log to display information from the Resources table, Workcell column that is associated with the task. Also, the Area Expression column was added to provide a different 'category' for grouping tasks into sections within the Gantt. These and additional examples are also discussed in the Entity Workflow Gantt help.



Used for the Task Row property

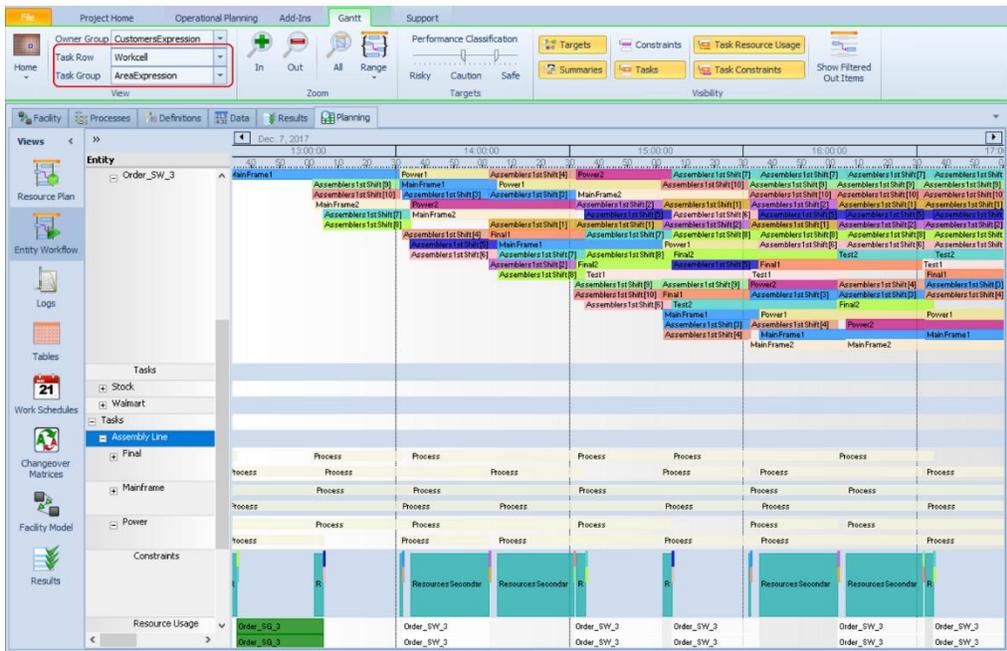


Used for the Task Group property

With Simio Sprint 162, we have enhanced both the Entity Workflow Gantt and Resource Plan Gantt ribbons to allow tasks to be displayed in different ways. See below Task Row and Task Group, circled in red, on the Gantt ribbon.

If only the Task Row references a Task Log column, the tasks are grouped by that column but still displayed directly under the entity that incurred the task (this property itself was added in Sprint 161). When a Task Group is specified (as below AreaExpression), all tasks are then *moved* from previously under each specific entity to more general Tasks area to allow for categorization of tasks. Tasks can then be shown by a location or area within a manufacturing facility (or any other custom expression specified). We have also enhanced the visibility of the Gantt items to include the Task Resource Usage and Task Constraints so that these may be turned on/off.

These new Gantt features will allow large assembly type operations with many tasks, like this example SchedulingBicycleAssembly, to be more clearly shown and customized on both the Resource Plan and Entity Workflow Gantt.

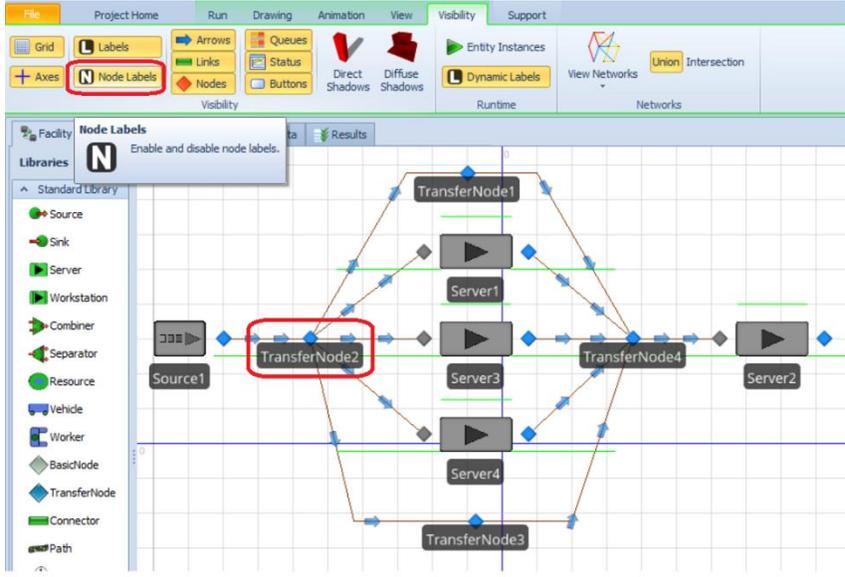


Using the Task Group option for categorizing tasks actually takes the Tasks out from under the particular order number (or entity) and puts them in a Tasks section. This allows the tasks for all orders (entities) to be seen in the same area.

If the Task Row is designated, the tasks will be categorized for each Task Group.

Node Name Visibility – Stand Alone Nodes

We have changed the software so that node object labels may now behave like other object labels in terms of visibility. Specifically, for “un-tethered” nodes (those not part of another object), we now show the name labels (previously labels would appear only when nodes were selected). This can be turned off within the Visibility ribbon.



New Process Functions

We have added a series of tokens in process-related functions. Note that these then provide access to the token reference that can then be used within the new token / task functions listed in the next section.

TokensInProcess.NumberItems - Returns the number of tokens that are currently executing the process.

TokensInProcess.FirstItem - Returns a reference to the first token in the list of tokens that are currently executing the process.

TokensInProcess.LastItem - Returns a reference to the last token in the list of tokens that are currently executing the process.

TokensInProcess.IndexOfItem(token) - Returns the one-based index of a specified token in the list of tokens that are currently executing the process. If the token is not found then the value 0 is returned.

TokensInProcess.ItemAtIndex(index) - Returns a reference to the token at a specified index position in the list of tokens that are currently executing the process.

TokensInProcess.Contains(token) - Returns True (1) if the list of tokens that are currently executing the process contains the specified token. Otherwise, the value False (0) is returned.

New Token.Task Functions

We have replaced the previous TaskInfo.* functions with Task.* functions and have added a number of new token/task related functions as well. These provide information related to a token within a task of the Task Sequences functionality. Thus, they are referenced as Token.Task, Token.Task.AssociatedObject, etc.

Task - Returns a reference to the active task associated with the token, if the token's execution is part of a task sequence.

Task.AssociatedObject - Returns the associated object reference for the task.

Task.SequenceNumber - Returns a string representing the sequence number used to determine the task's precedence constraints.

Task.IDNumber - Returns the integer number used to identify the task in the Immediate Predecessors or Immediate Successors field of another task.

Task.Name - Returns the name for the task.

Task.ExecutionID - Returns the unique integer identifier number automatically assigned to the task when the StartTasks step was executed.

Task.TimeStarted - Returns the simulation time (in hours) that the task was started.

Task.TimeInProgress - Returns the elapsed time duration (in hours) since the task was started.

Task.SeizedResources.NumberItems - Returns the number of resources currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution.

Task.SeizedResources.FirstItem - Returns a reference to the first resource in the list of resources currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution.

Task.SeizedResources.LastItem - Returns a reference to the last resource in the list of resources currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution.

Task.SeizedResources.IndexOfItem(resource) - Returns the one-based index of the first occurrence of a specified resource in the list of resources currently seized by the task's associated object, filtered to only

include the resource seizes that occurred specifically due to the task's execution. If the resource is not found then the value 0 is returned.

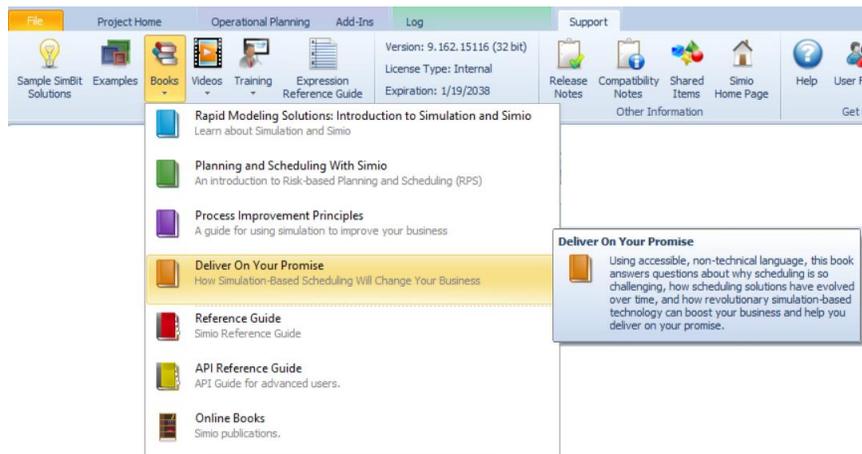
Task.SeizedResources.ItemAtIndex(index) - Returns a reference to the resource at a specified index position in the list of resources currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution.

Task.SeizedResources.Contains(resource) - Returns True (1) if the specified resource is in the list of resources currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution. Otherwise, the value False (0) is returned.

Task.SeizedResources.CapacitySeizedOf(resource) - Returns the total number of capacity units of a specified resource that are currently seized by the task's associated object, filtered to only include the resource seizes that occurred specifically due to the task's execution.

Support Ribbon - Books

We have added a link to a new e-book, *Deliver on Your Promise*, written by Simio CEO, C. Dennis Pegden, PhD. This book discusses why scheduling is so challenging, how scheduling solutions have evolved and how simulation-based technology can boost your business.



Simio Update to .NET 4.6.2

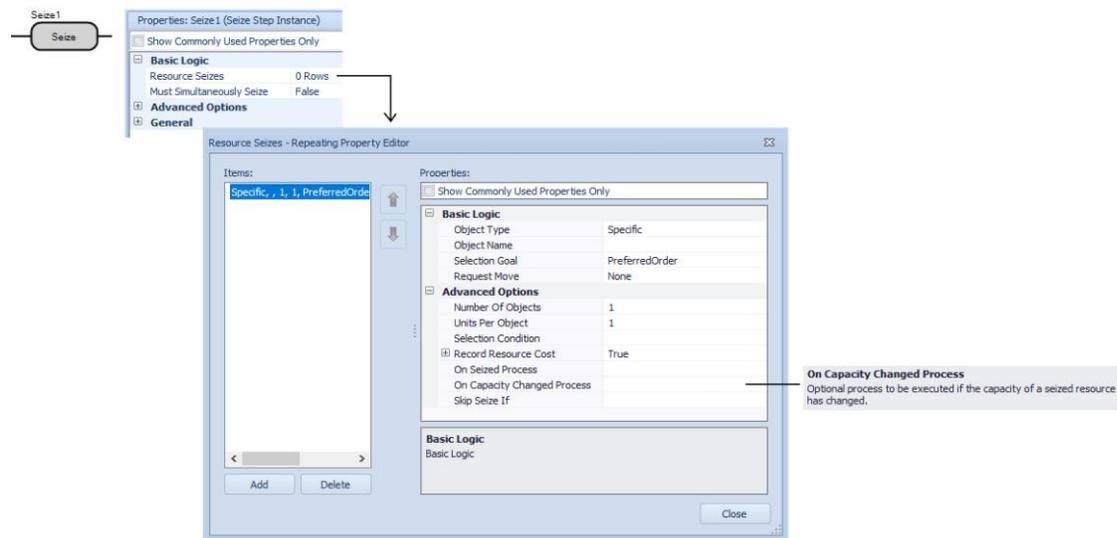
Simio requires the *.NET Framework Version 4.6.2*. If the framework is not already installed on your computer and you want to perform a manual install, you can download it from the *.NET Framework Downloads* page: (<http://msdn2.microsoft.com/en-us/netframework/aa569263.aspx>). It may also be installed by doing a Windows Update, depending upon your particular system configuration.

Simio Release 9 – Sprint 160/161 – September 2, 2017

In this sprint, we have made several changes in conjunction with some more major enhancements to come in the next several sprints. First, the Seize step has been enhanced to allow an additional process when the capacity of the seized object changes. Second, a unique Task Id field has been added to the Logs within the Planning tab (Simio Enterprise) that will allow for more customized Gantt viewing in the future.

Seize Step Enhancement – On Capacity Changed Process

We have enhanced the Seize step to include an On Capacity Changed Process property to optionally specify a process to execute if the seized resource capacity changes.



When a seized resource object's capacity has changed, the *On Capacity Changed Process* for each of the resource's owners (if any were specified at the Seize step(s)) will be first executed in seize order. Then last the resource object's *OnCapacityChanged* process.

When a new token is created to execute the *On Capacity Changed Process* that was specified at a Seize step:

- The token's associated object reference will be the resource.
- The token's context object reference will be the resource owner.
- The token will have a reference to the same active Task (if applicable) as the token that executed the Seize step.
- The token will have the same table references as the token that executed the Seize step.

Suspend and Resume Step Enhancements

We have enhanced both the Suspend and Resume steps to include *Token Match Condition* and *Skip Suspend If* properties. The *Token Match Condition* properties are available when the *Suspend Type* or *Resume Type* are 'Process'. These are optional match conditions used to filter the tokens executing the process. Only tokens currently in process that satisfy this condition will be suspended. The *Skip Suspend If* property is located within the Advanced Properties section of the repeatable property editor for both

the Suspend and Resume steps. This optional condition indicates whether to skip the suspend or resume action.

New Resource Object Function

ResourceOwners.AllProcessingSuspended - Returns True (1) if each owner of the resource has at least one associated process token currently suspended. Otherwise, the value False (0) is returned.

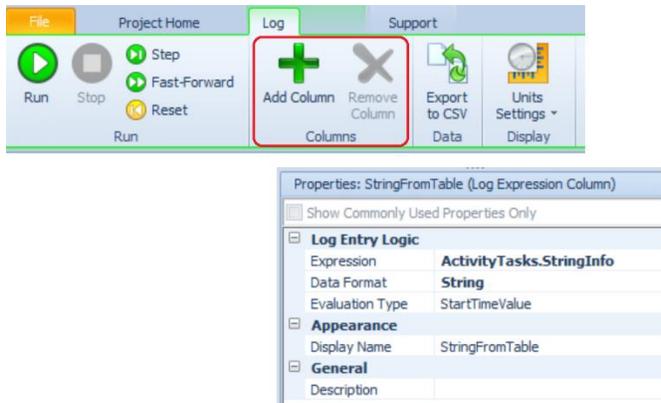
Simio Enterprise Edition – Task Id in Logs, Add Column Expressions within Task Log

We have enhanced many of the logs within Simio Enterprise Edition so that users may customize the Gantt charts.

Within the Task Log, there is a new Task Id column which includes a unique identifier for each task that was incurred. The Task Id may not always be sequentially listed, as the task is assigned a unique Task Id before the task is performed. If for some reason the task is not completed, you will not see it in the log.

Then, within many of the other usage and constraint logs, the Task Id is displayed. Within the Resource Usage Log, Material Usage Log, Transporter Usage Log and Constraint Log, the Task Id associated with the resource, material, transporter, or general constraint will be displayed.

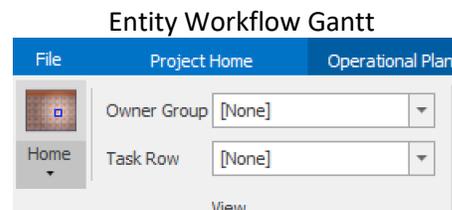
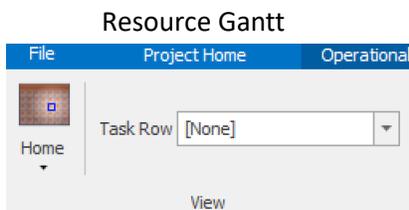
In addition to the Task Id available in the Task Log, we have added the ability to add/remove an expression type column(s) to the Task Log. The Log ribbon includes both the Add Column and Remove Column buttons (similar to adding new expression columns to the Resource Usage Log).



Note that the token used to evaluate the expression is the one created to run the process associated with the task, so whatever expression you would use in the model to get information, you would use here as well (e.g., Entity.** for the associated object that was indicated for StartTask, Token.TaskInfo.** for info about the task itself, TableName.ColumnName for table info associated with the running task).

Simio Enterprise Edition – Resource Plan and Entity Workflow Gantt - Task Row combo box selection

We have added a Task Row selection box to both the Resource Plan and Entity Workflow Gantt, where you can specify which custom expression on the Task Log is used to indicate what row a task item goes in. If nothing is specified, then the tasks continue to go into the “Tasks” row under the corresponding resource/owner. If a task row is specified, then the tasks go in a row of that name ***under*** the existing “Tasks” row (expanded with the +). Note that the original Grouping Expression for the Entity Workflow Gantt is now named Owner Group.



Task Id	Entity Id	Entity	Facility Location	Station	Task Sequence	Task	Start Time	End Time	Duration (Hours)	Location
1044	MFGOrder.696	Order_SG_1	MainFrame2	Processing	ProcessingTaskSequence	Process	12/1/2015 8:00:06 AM	12/1/2015 8:06:25 AM	0.105438	MainFrame2
1047	MFGOrder.696	Order_SG_1	MainFrame1	Processing	ProcessingTaskSequence	Process	12/1/2015 8:00:06 AM	12/1/2015 8:06:26 AM	0.105519	MainFrame1
1049	MFGOrder.696	Order_SG_1	Power2	Processing	ProcessingTaskSequence	Process	12/1/2015 8:06:28 AM	12/1/2015 8:13:42 AM	0.120513	Power2
1050	MFGOrder.695	Order_SG_1	Power1	Processing	ProcessingTaskSequence	Process	12/1/2015 8:06:29 AM	12/1/2015 8:13:44 AM	0.120723	Power1
1052	MFGOrder.697	Order_SG_1	MainFrame2	Processing	ProcessingTaskSequence	Process	12/1/2015 8:06:29 AM	12/1/2015 8:12:33 AM	0.101046	MainFrame2
1055	MFGOrder.698	Order_SG_1	MainFrame1	Processing	ProcessingTaskSequence	Process	12/1/2015 8:06:30 AM	12/1/2015 8:12:50 AM	0.105519	MainFrame1
1058	MFGOrder.699	Order_SG_1	MainFrame2	Processing	ProcessingTaskSequence	Process	12/1/2015 8:12:37 AM	12/1/2015 8:18:41 AM	0.101133	MainFrame2

Navigation: Model

- SchedulingBicycleAssembly
 - ModeEntity
 - Model

Properties: Location (Log Expression Column)

Show Commonly Used Properties Only

Log Entry Logic

Expression: **Entity.Location.Parent**

Data Format: **String**

Evaluation Type: **StartTimeValue**

Appearance

Display Name: **Location**

General

Project Home | Operational Planning | Add-Ins | Gantt | Support

Owner Group: Customer

Task Row: **Location**

View: In | Out | All | Range

Performance Classification: Risky | Caution | Safe

Targets | Constraints | Summaries | Tasks | Show Filtered Out Items | Visibility

Views: Entity

- Tasks
 - Final1
 - Final2
 - MainFrame1
 - MainFrame2
 - Power1
 - Power2
 - Test1
 - Test2

Entity Workflow

Logs

Dec. 4, 2015

09:00:00

Process | Process | Process | Process

10 | 10 | 10 | 10

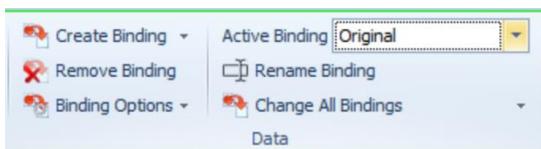
Process

Simio Release 9 – Sprint 159 – July 24, 2017

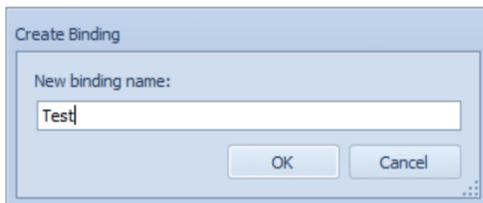
In this sprint, we have made many enhancements to the Data Binding capabilities within data tables. This includes having multiple bindings per table, changing between bindings for multiple tables easily and experiment scenario based data binding. Additionally, we have added the ability to remove rows from an output table and have included 2 SimBits demonstrating lists in tables.

Data Table Binding Enhancements

We have enhanced the data binding capabilities for data tables, including adding an *Active Binding* selection to allow for multiple bindings per data table. The original binding for a data table is considered the 'Default' binding (*Active Binding* is 'Default'). The *Rename Binding* option allows users to change the 'Default' name to another name (such as 'Original' as shown below).



When adding a new binding, use the *Create Binding* option. If the table already has an associated binding, the Create Binding dialog will appear asking for the new binding name. This binding name can be the same across multiple tables.



Each table then has its own 'group' of bindings. The *Change All Bindings* option then allows the user to set the Active Binding for several tables at one time. If there are multiple tables with the same binding name, they will be displayed in the *Change All Bindings* pull down list.



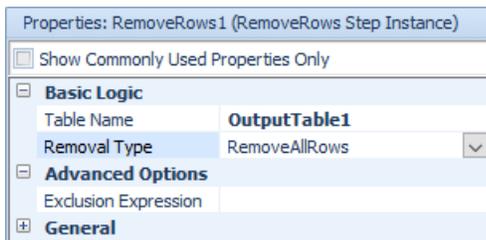
Experimentation – Scenario Based Data Binding

We have a new set of columns in the Design view of Experiments. Each column corresponds to every table with two (2) or more data bindings. The headers of the columns are the table names. The cell values are a drop down of the named bindings for that table. The default value of the cell for a new scenario is the currently 'active' data binding for that table. Then, for a particular scenario, the user can set the various bindings desired per table. OptQuest scenarios will keep data bindings as the 'active' data binding for the table.

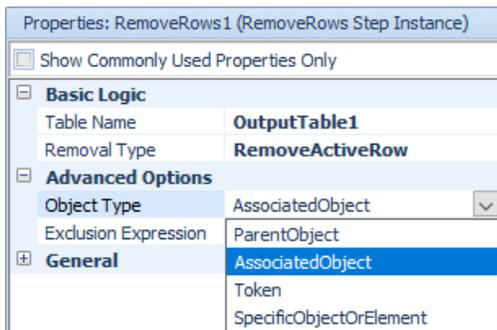
Design Response Results Pivot Grid Reports Input Analysis										
Scenario			Replications		Controls		Table Bindings		Responses	
<input checked="" type="checkbox"/>	Name	Status	Required	Completed	NumberOfDoctors	NumberOfNurses	PatientPriority	Initialization	Cost Of Staff	Average Length Of Stay
<input checked="" type="checkbox"/>	001	Idle	5	5 of 5	3	3	Original	Original	31247.5	34.976
<input checked="" type="checkbox"/>	002	Idle	10	10 of 10	3	3	Test	Original	31273.8	34.3466
<input checked="" type="checkbox"/>	Scen...	Idle	10	10 of 10	3	3	Original	Test	31394.1	27.3299
<input checked="" type="checkbox"/>	Scen...	Idle	10	10 of 10	3	3	Test	Test	31394.1	27.3299

RemoveRows Step Enhancement

The RemoveRows step now can remove a single row from an output table. A new *Removal Type* property defaults to the step's existing behavior:



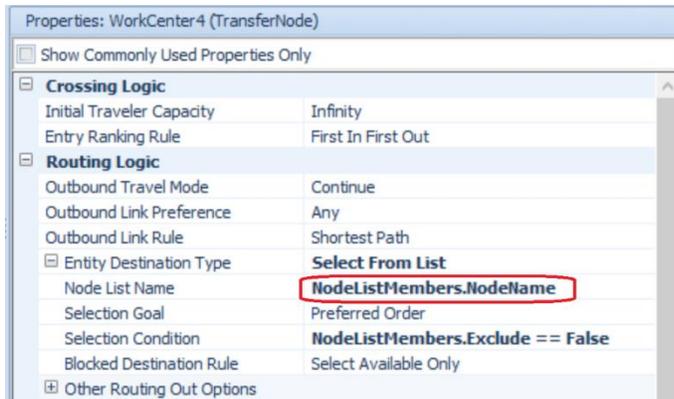
Setting the *Removal Type* to 'RemoveActiveRow' allows users to specify which object or element has a reference to the row to be removed:



This means that to remove a row from an output table, you use a SetRow step followed by a RemoveRows step. Once a row has been removed, any object or element that had a reference to the removed row now has an *invalid* reference, and it must be reset to a valid row before attempting to use it.

List Members in Tables – Enhancement to Destination Node List Name

We have added functionality so that users can now specify the *Node List Name* for a TransferNode (or *Destination Node List Name* for a Routing Group element) as a column in a data table.



New SimBits

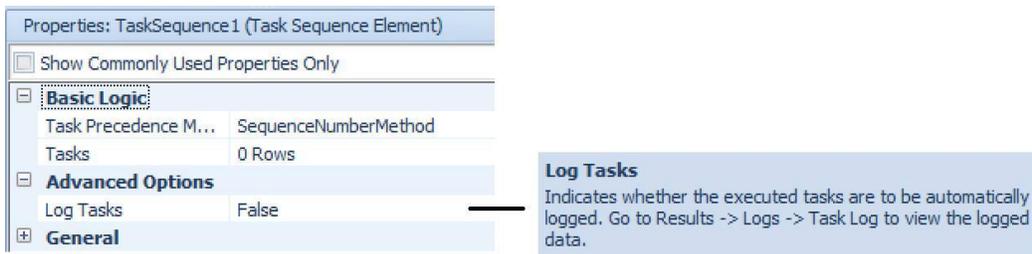
We have added a two new SimBits to demonstrate the use of tables for defining lists of objects.

Using Relational Tables To Define Node Lists – In this model, entities are processed with different routings based on type, and each job step in a routing is performed at a work center that consists of two or more identical servers. All product mix, job routing, lists of possible server input nodes for each job entering a work center, and operation data is defined in a set of relational data tables.

Using Relational Tables To Define Resource Task Lists – This model includes a server where the entity processing is a task sequence, and where all operation data is defined in a set of relational data tables including lists of possible resources that can perform specific tasks.

Task Logging

The TaskSequence element now provides a new *Log Tasks* Boolean property in Advanced Options.



And now for the Server, Combiner, or Separator objects in the Standard Library, the *Log Resource Usage* property provided in Advanced Options of each of those objects is now mapped down to the *Log Tasks* property of its child 'ProcessingTasks' TaskSequence element.

Simio Release 9 – Sprint 157/158 – June 30, 2017

In this sprint, we have expanded the type of characters that may be used within identifier names of objects and elements. Additionally, we have enhanced the route selection algorithm for the RoutingGroup element to better support common scheduling scenarios.

New Identifier Rules - Expanded

Identifiers, such as names of objects within the Facility window and elements within the Definitions window, can now start with a number and/or use expanded character set. For example, '123abc' is now a valid object/element name. We also have greatly expanded the available characters that are valid, so all Unicode uppercase and lowercase letter categories and well as others are supported. (As an example, here is a listing of the Unicode lower case letter category <http://www.fileformat.info/info/unicode/category/Ll/list.htm>)

Simio doesn't allow identifiers that evaluate to numbers so '123' is not valid, but '3f4' is valid. '3e4' is not valid because that is exponential notation for 3×10^4 .

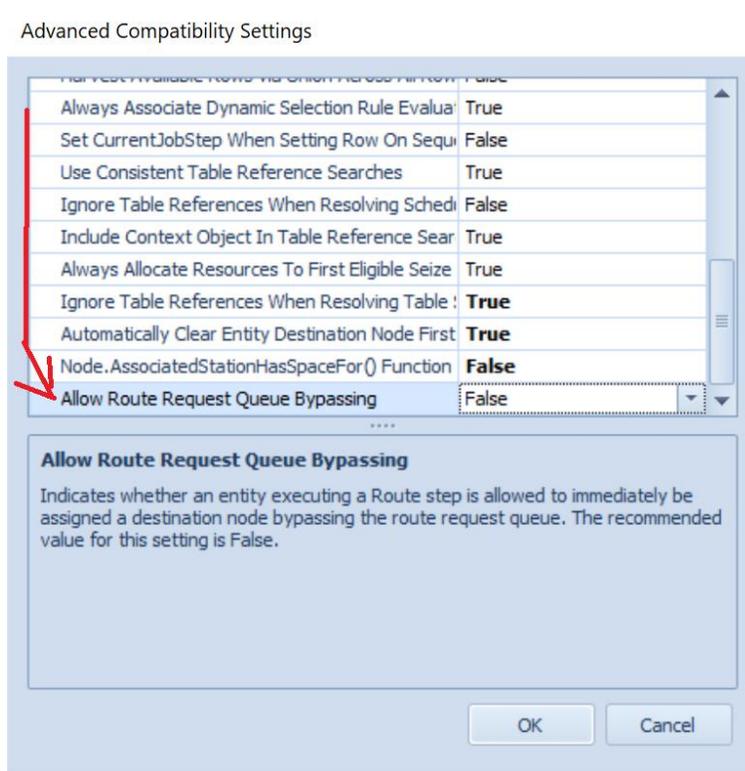
Routing Group – Route Selection Algorithm Enhancements

The RoutingGroup element has been enhanced so that the route selection algorithm prevents a lower priority entity from bypassing a route request queue that potentially contains higher priority entities. Additionally, if multiple destination candidates available in the list of possible destination nodes, the route selection algorithm consistently applies the destination selection goal specified on the Route step or TransferNode.

More specifically, there are two changes that have been made to the route selection algorithm in sprint 157.

- 1) When a Route step is executed, the arriving entity will no longer try to immediately select a destination. Instead, the entity will always be placed into the route request queue. The event types that trigger a routing group to schedule a late priority current event to try to select entities from its route request queue will now include any new entity arrival. See RoutingGroup element help for more information on current/new selection algorithms.

In the Run tab of the Ribbon, in Advanced Compatibility Settings, there will be a new Allow Route Request Queue Bypassing setting. The default value for this setting will be True for old models and False for new models.



2) When checking its route request queue, the new selection algorithm used by the routing group will be as follows:

If no dynamic selection rule is being used (static queue ranking rule only)

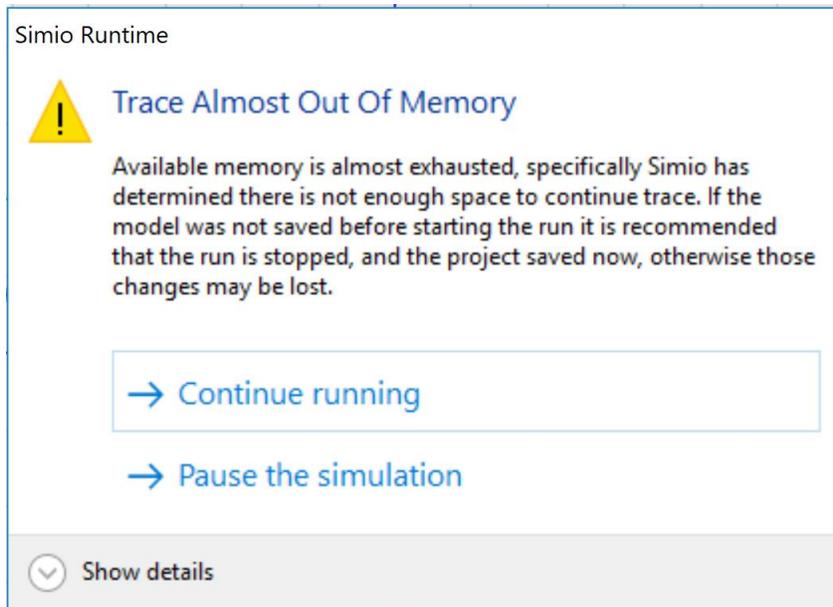
- Use the route request queue's static ranking rule to select a queue item.
- If there are multiple possible destination nodes available for the selected queue item, then use the entity's destination selection goal to select one of those nodes.

If dynamic selection rule is being used

- Find the best route request queue item for each possible destination node using the specified dynamic selection rule.
- Use the route request queue's static ranking rule to select a queue item from the best queue items found.

Memory Check for Trace

If trace is on, and Simio detects during trace that there is less than 50 MB of memory left, it now stops tracing and give you a warning, allowing for a model save:



Command Line Option for *Start in Scheduling Mode*

We have added a new command line option for Simio Enterprise users who may be giving a demonstration to production schedulers. The new "-start-in-scheduling-mode" command line option will simplify the interface and focus on the Gantt, Dashboards and Reports. For example, a shortcut for opening one of the scheduling examples would be:

"C:\Program Files (x86)\Simio\Simio.exe"

"C:\Users\Public\Documents\Simio\Examples\SchedulingDiscretePartProduction.spfx" -start-in-scheduling-mode

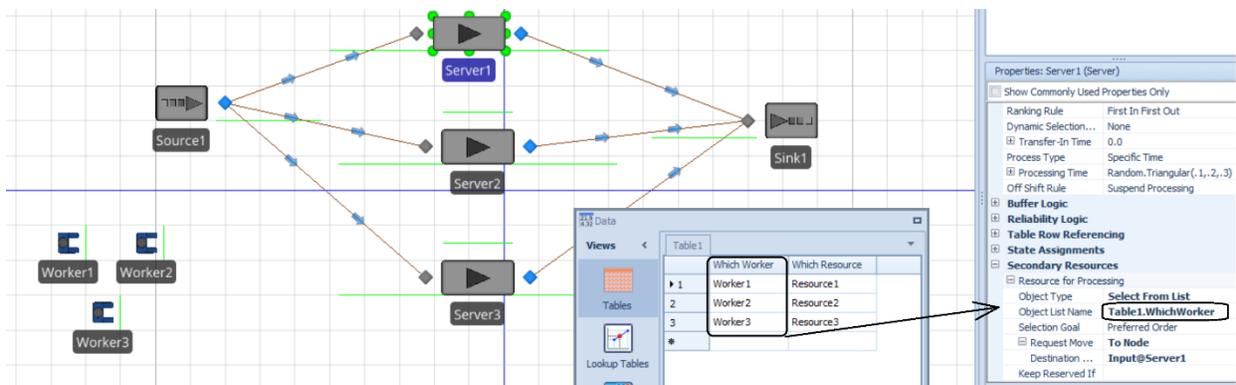
Simio Release 9 – Sprint 156 – June 2, 2017

In this sprint, we have added the capability to specify and reference certain object lists from tables, as well as enhanced our Gantt support zooming. We now also save the xml files for each object definition and table separately when saving *.simproj files to allow for easier concurrent modeling of projects.

List Members in Tables – Enhancement to Seize/Move/Release

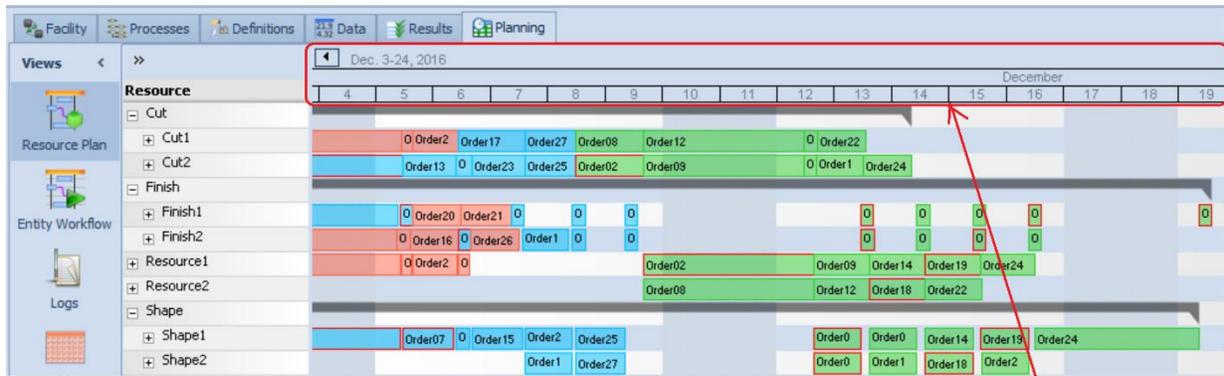
We have added the capability within several steps to allow for list members to also be specified within tables (in addition to Lists). The Seize, Move and Release steps now all allow for the TableName.ColumnName reference to be used as the *Object List Name* property value. Note that this table reference does not currently appear on the pull-down list for that property. Given that the Standard Library objects use the Seize/Release steps for allocating secondary resources, these table references are also available within the Secondary Resources section of objects (as shown below), as well as the Resource Requirements section within Task Sequence type processing.

Please note that this also includes the use of table state columns (Enterprise edition) for the list which, combined with the Output tables and the RemoveRows step, allows for dynamic runtime building of Seize lists.



Gantt Scroll Wheel Support – Enterprise Edition

We have enhanced the Simio Enterprise Edition Gantt charts so that the scroll wheel can be used to zoom in/out. Within either the Resource Gantt or Entity Flow Gantt charts, the cursor should be placed within the time scaler area to enable the scroll wheel for zooming. This now functions similarly to the Simio Portal edition for the Gantt charts.



Place cursor in the time scaler area and use scroll wheel to zoom in and out

New SimBit

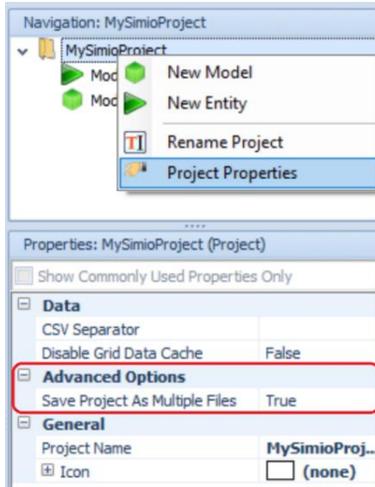
We have added a new renegeing type SimBit to our Sample SimBit Solutions.

RenegingCombiner – this model illustrates a palletizer or similar machine (Combiner object) that batches a standard number of parts together. After the parent (pallet) entity has waited for a given amount of time for the individual parts to arrive, it will renege, reset the batch size to the number of member parts currently waiting and then re-enter the combiner to continue with a smaller batch size.

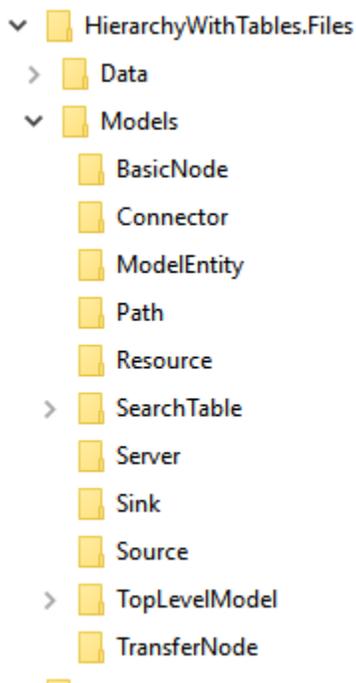
Enhancement to Allow for Easier Concurrent Modeling

Simio Team and Simio Enterprise licenses no longer check for an overall signature on the entire file. This will enable merges in various source control scenarios to generate still readable files by the user modelers. Note that Simio added an additional signature on the “document info” section of the file which must be present for the file to load if the signature of the entire file is missing or incorrect. Thus, if a user does wish to use this functionality, they will need at least one save in the new version of Simio to generate this document info signature.

For multi-file save targets (simproj and the internals of the zip archive of the sfx), Simio will save each Object Definition to its own file and each Table to its own file ***IF*** the project level property setting of *Save Project As Multiple Files* property is set to ‘True’ (which is the default for new projects). If it’s set to ‘False’ (the setting for all existing projects), we will continue saving the way we currently do (thus keeping existing simproj files “looking” the same when they are saved with new software).



To make use of the multi-file feature, the user should save the project as a .simproj. When the model is saved with this target, each object in the model will have its information stored in a folder by the same name located in the Model.Files\Models folder. For example, the HierarchyWithTables SimBit would look as follows:



Each folder will contain an XML file with information pertaining to that object type. If multiple modelers are working on the same project, but different objects, the updated object folder can be copied over the existing object folder. In the above example, if a modeler wanted to make changes only to the SearchTable object, the modeler can make the edits in a separate copy of the model, save the model as a *.simproj and then replace the old SearchTable folder with the updated folder.

Academic Licensing

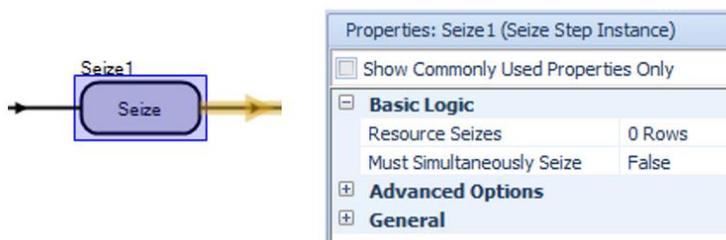
Please Note: In this and future Simio versions, projects saved with an academic license will no longer have their contents written in human-readable XML.

Simio Release 9 – Sprint 155 – May 1, 2017

In this sprint, we have enhanced the Seize step (and corresponding seize operations within the Standard and Flow Library objects) to allow users to **simultaneously seize** multiple resources. We have also enhanced the Seize trace to provide more information when seizing resources.

Seize Step – Simultaneous Seizing Enhancement

We have enhanced the Seize step such that if multiple resources are required, there is an option that indicates whether all must be available before any can be seized. If *Must Simultaneously Seize* is set to 'True', all resources must be available before they are seized. For more information, refer to the Seize step Help page within Simio.



Seize Step – Enhanced Trace

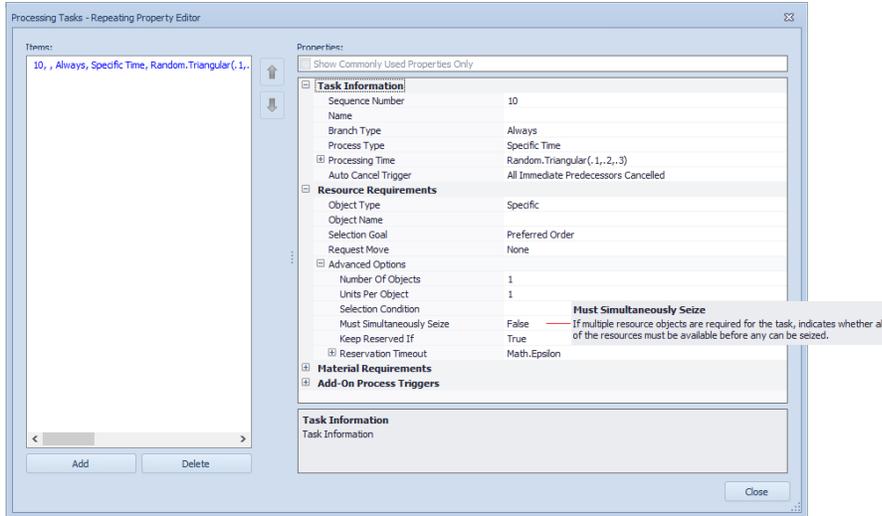
In addition to the above Seize step changes, we have also enhanced the Simio 'trace' for the step so that users may more easily view why a resource or group of resources is seized or not seized. In the below screenshot, for example, the entity requires both Tool3 and Worker1. While Tool3 is available for seizing at the time, Worker1 is currently not available. Therefore, the trace indicates that 'Insufficient resources available to satisfy the Must Simultaneously Seize requirement'. The entity will wait for both those to be available at the same time before seizing both.

Stopped at breakpoint on Process 1.Seize2	
[Seize] Seize2	Object 'DefaultEntity.34' checking availability to seize '1' resource(s) of resource type '[Specific] Tool3'.
	Object 'DefaultEntity.34' found '1' currently available resource(s) of resource type '[Specific] Tool3'.
	Object 'DefaultEntity.34' checking availability to seize '1' resource(s) of resource type '[Specific] Worker1'.
	Object 'DefaultEntity.34' unable to seize '1' capacity unit(s) of resource 'Worker 1[1]'. Insufficient capacity available.
	Object 'DefaultEntity.34' found '0' currently available resource(s) of resource type '[Specific] Worker1'.
	Object 'DefaultEntity.34' unable to seize resources. Insufficient resources available to satisfy Must Simultaneously Seize requirement.
	Object 'DefaultEntity.34' waiting to seize '1' resource(s) of resource type '[Specific] Tool3'.
	Object 'DefaultEntity.34' waiting to seize '1' resource(s) of resource type '[Specific] Worker1'.

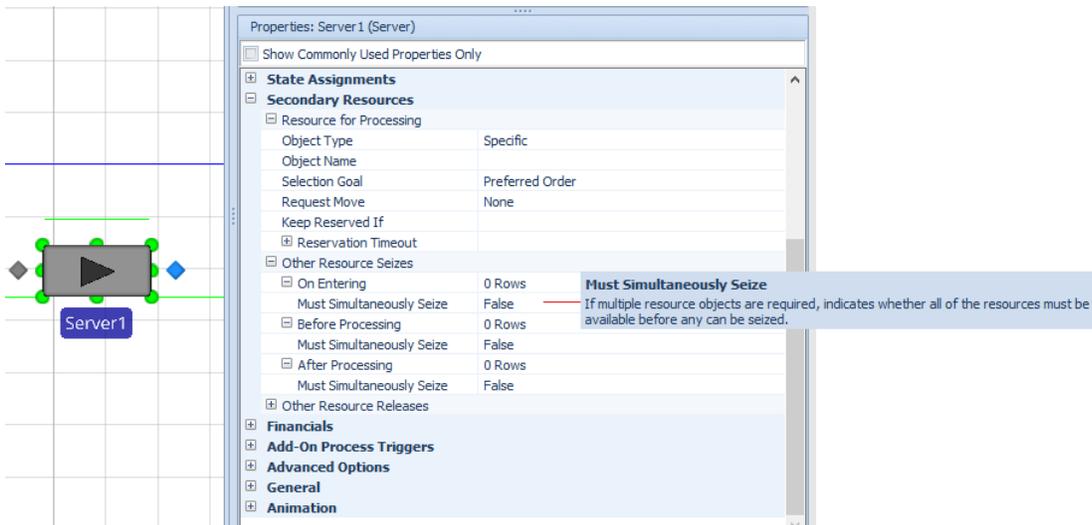
Standard and Flow Library Enhancements – Server, Combiner, Separator, Filler, Emptier

Based on the above enhancement to the Seize step, we have incorporated that same feature into our library object where multiple task resources or secondary resources may be utilized.

For task sequence type processing, processing tasks, the *Must Simultaneously Seize* is specified within the Processing Tasks repeating property. If the resources are specified and referenced within a table or if multiple objects are required as specified by the Number of Objects and/or Units per Object, this new field will be used.



For secondary resources within processing of the various objects, the 'Other Resource Seizes' sections of properties each have the option for *Must Simultaneously Seize*.

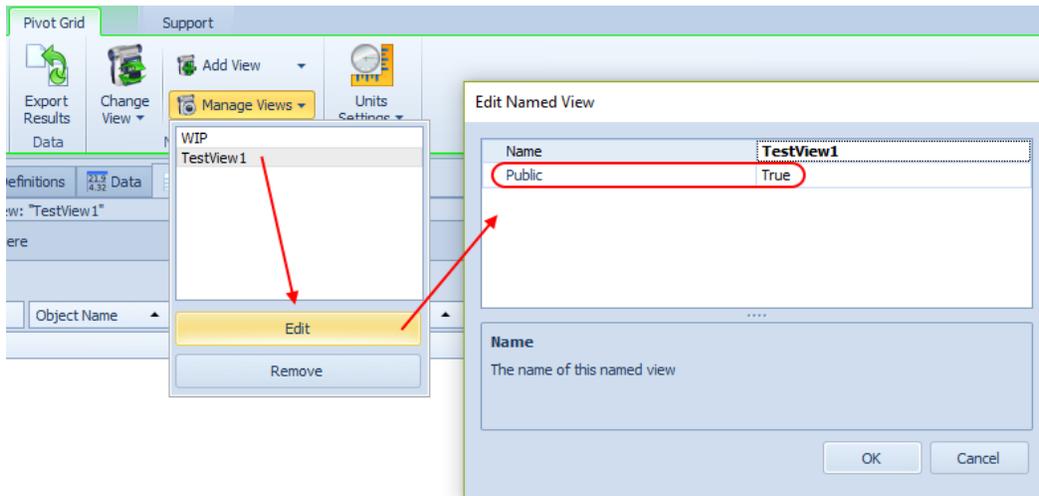


Simio Release 9 – Sprint 154 – April 10, 2017

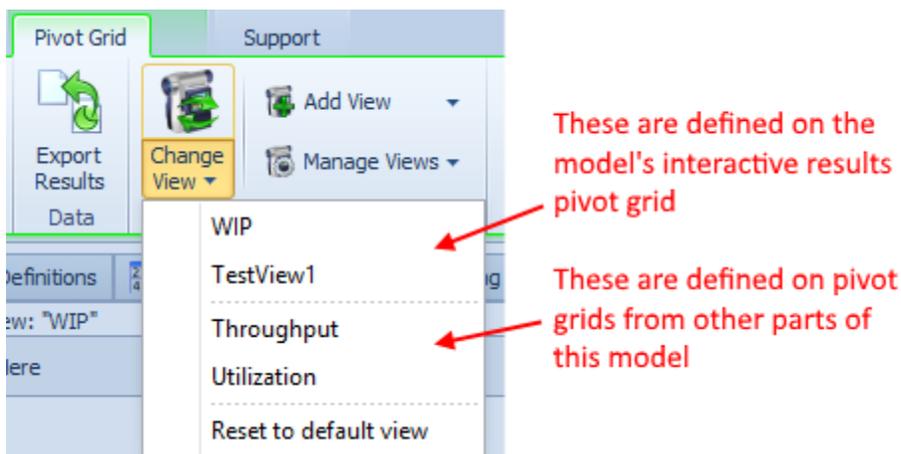
In this sprint, we have added the much-requested feature of *public* views for the pivot grids in different areas of the product. This allows users to customize pivot grid reports in a particular window (i.e., interactive results window), specify them as 'Public' and then view those same customized views in other windows, such as experiment pivot grid or planning results detailed results pivot grid.

Pivot Grid – Public Views

Named views in our various pivot grids are now "Public" by default, allowing them to be used by any other pivot grid in the model or any of its experiments:



When using named views in a pivot grid, you'll now see public ones from the other pivot grids associated with the same model. For example, when viewing the Results pivot grid for a model, you'll see its own named views, followed by public named views from other pivot grids (including those defined by the model's Experiments, as well as Planning results in Enterprise edition).

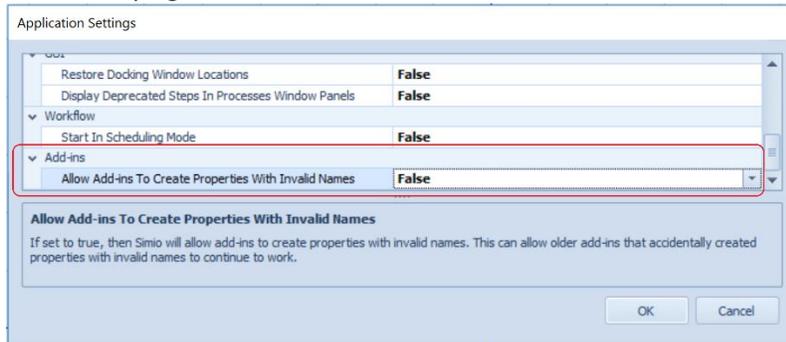


When applying a named view to a pivot grid, we will automatically hide or show certain columns as appropriate to the pivot grid. Specifically, the Minimum, Maximum, Half Width, and Std. Dev. columns will be hidden when applying an experiment's named pivot view to a model's pivot grid, while the

Minimum, Maximum, and Half Width columns will automatically be shown when applying a named pivot view from a model's pivot grid to an experiment's pivot grid.

New Application Setting - Allow Add-ins To Create Properties With Invalid Names

There is a new application setting that lets users turn off the validity checking of names of properties created by the API. The reason this has been added is that we did not used to check for validity, and there are old add-ins that create improper property names (e.g., with spaces) which will now cause models relying on those add-ins to not load in some circumstances.



Poisson Distribution

Please note that with the Poisson distribution for means ≥ 40 , the distribution will be calculated using the Normal distribution.

Simio Release 9 – Sprint 153 – March 21, 2017

In this sprint, we have added some new work schedule related functions, as well as enhanced our table importing capabilities. Several user-requested features have also been added, including a Sequence Destination property type for changing a data table to a sequence table and an Entity Activity List report for Simio Enterprise users.

New Schedule Functions

The below schedule based functions have been added to allow users more flexibility accessing schedule based information. These functions can be accessed using ScheduleName.*, SchedulePropertyName.Schedule.* (e.g., ResourceObjectName.WorkSchedule.Schedule.* if the resource is following a work schedule) or TableName.SchedulePropertyName.Schedule.*

AverageValue(fromDateTime, toDateTime) – Returns the average value over a specified time range.

MinimumValue(fromDateTime, toDateTime) – Returns the minimum value over a specified time range.

MaximumValue(fromDateTime, toDateTime) – Returns the maximum value over a specified time range.

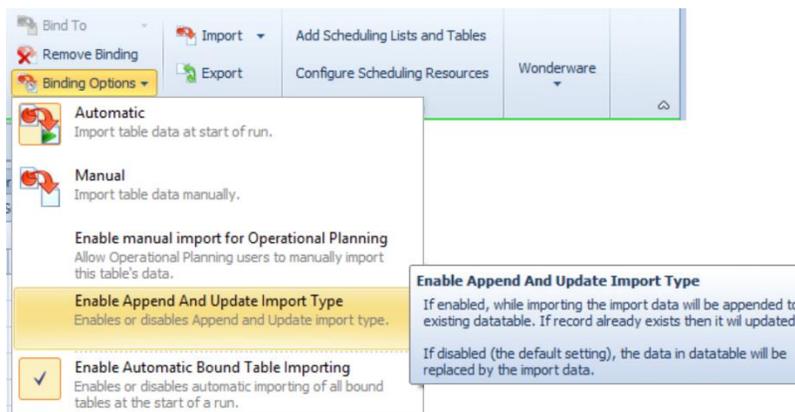
New SimBit

We have added a new SimBit to our large selection of small simulation modeling examples.

VehicleDoingSimultaneousLoading – this model illustrates three entity types waiting to be picked up by a vehicle at the same time. Each entity type has a different load time and the vehicle departs after the last entity finishes loading. The logic in this model could be used to customize the Vehicle object.

Table Importing Enhancement

The Binding Options within Data tab / Tables now includes the option to append and update data when importing. If this option is enabled and data is imported into a table with existing data, any new data will be appended to the existing table. If a record already exists (based on a value in the key column of that table), the data will be updated. * Note: This feature is available within sprint 152, but was not documented until sprint 153.



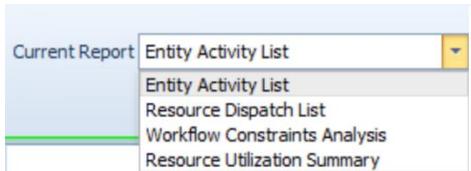
Tables - Sequence Destination Property

A standard Data Table can now be changed into a Sequence Table by using the new 'Sequence Destination' property type. This can be done by adding a new column of that type or changing an

existing column to that type. This is useful if the user has already defined multiple columns in a standard Data Table and realizes it should be a Sequence Table OR if the sequence destination column of a Sequence Table has been accidentally deleted.

New Entity Activity List – Simio Enterprise

Within the Planning tab, Results panel for Simio Enterprise edition, we have added an additional report named Entity Activity List. This report is an entity based listing of all resources / activities that have been allocated over time. The *Entity Name*, as well as *Start Date* and *End Date* are specified. As with the other reports in this section, the 'Show Detail' option provides more details on a resource by resource basis.



A screenshot of the Simio Enterprise Results panel. The 'Entity Activity List' report is displayed. The interface includes a left sidebar with navigation options like 'Resource Plan', 'Entity Workflow', 'Logs', 'Tables', 'Work Schedules', 'Changeover Matrices', 'Facility Model', and 'Results'. The main area shows the report details and a table of resource activities.

Entity Activity List
 Generated On: 3/21/2017 8:12:24 AM
 Date Range: 12/1/2016 8:00:00 AM - 12/22/2016 8:00:00 AM [Show Detail](#)
 Owner: Order01

Resource	Start Time	EndTime	Duration (Hours)
Weld1	12/1/2016 8:00:17 AM	12/1/2016 11:42:17 AM	3.7000
Resource1	12/1/2016 8:00:17 AM	12/1/2016 11:42:17 AM	3.7000
Shape1	12/1/2016 11:42:30 AM	12/2/2016 8:30:30 AM	20.8000
Finish1	12/2/2016 8:30:41 AM	12/2/2016 1:54:41 PM	5.4000

Where the 2nd page is shown in graphical form (when Show Graphics is set to 'Yes')

Date Range: 12/1/2016 8:00:00 AM - 12/22/2016 8:00:00 AM [Show Detail](#)
 Owner: Order01

Weld1

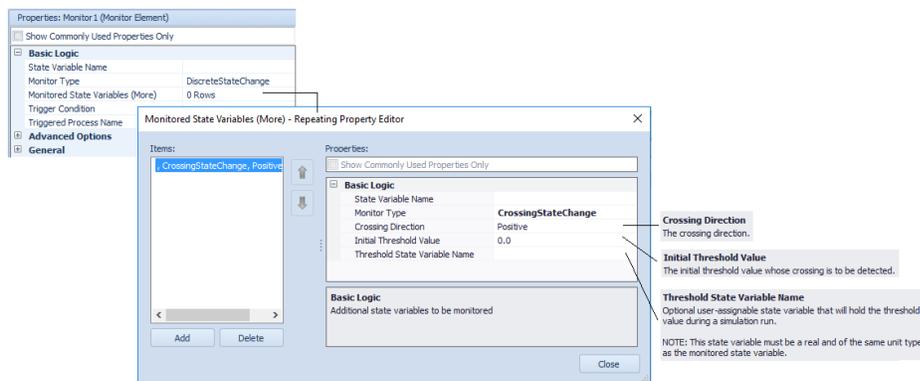
Simio Release 9 – Sprint 152 – February 24, 2017

In this sprint, we have updated the Monitor element to allow for multiple state variable triggers. A SimBit project has been added to demonstrate this feature in comparison with the new Scan step functionality that was added in Sprint 151. We have also added several user requested features needed for simulation projects.

Monitor Element Enhancement

We have enhanced the Monitor element to be able to monitor more than one state variable, which will simplify the event-driven waiting approach (Wait step). The enhancement will make it easier to model a single 'Status Changed' event that pertains to some logical grouping of state variables in the system, such as a set of variables that relate to the status of an area, a processing location, etc.

For example, suppose a single Monitor element named PackingAreaStatusChanged is defined that is monitoring discrete changes in the state variables Packer1.AllocationQueue, Packer1.ResourceState, Packer2.AllocationQueue, or Packer2.ResourceState. Somewhere in the modeled process logic, perhaps a Wait step is then used to hold a process token until event name PackingAreaStatusChanged occurs and event condition Packer1.ResourceState==0 && Packer2.ResourceState==0 is true.



New SimBit

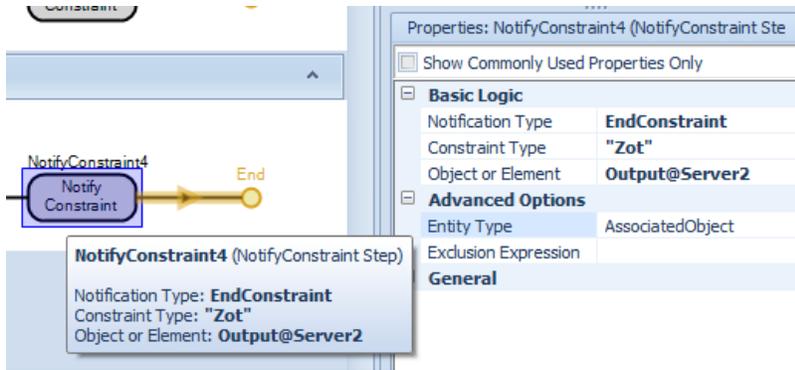
We have added a new SimBit project (BatchingProcessUsingScanOrWaitStepToControlBatchSize.spx) that includes two models which display the polled waiting (Scan step) approach and event driven (Wait step) approach.

BatchingProcessUsingScanStepApproach - Batching process at a server where the batch size control is modeled using a Scan step approach (polled waiting).

BatchingProcessUsingWaitStepApproach - Batching process at a server where the batch size control is modeled using a Wait step approach (event-driven waiting).

New NotifyConstraint Step

The new NotifyConstraint step allow users to add custom (i.e. user-defined) constraint entries into the Constraint Log (and by extension the Entity Gantt chart).



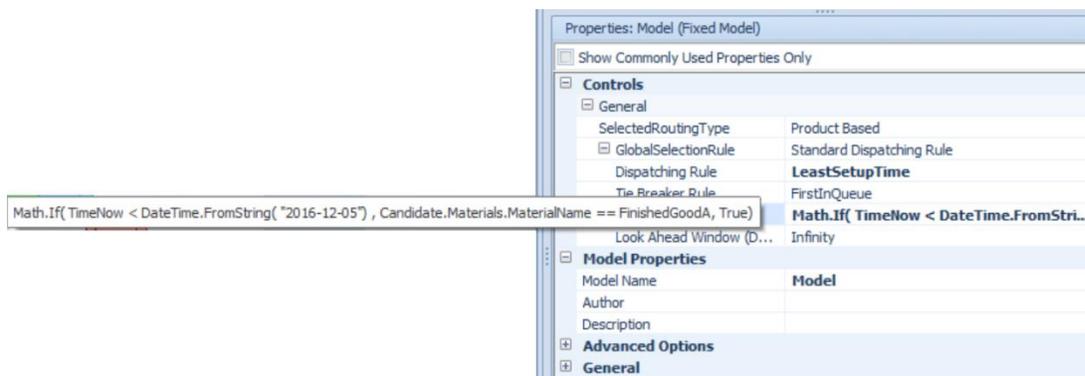
Entries in the Constraint log show up like this (the first two rows are existing automatic ones, the next two are custom ones):

Entity Id	Entity	Facility Location	Station	Constraint Type	Constraint Item Id	Constraint Item	Constraint Description	Start Time	End Time
DefaultEntity.33	DefaultEntity.33	Server1	InputBuffer	Resource Availability	Server1	Server1	Server	2/13/2017 12:04:33 AM	2/13/2017 9:03:39 AM
DefaultEntity.34	DefaultEntity.34	Server1	InputBuffer	Resource Availability	Server1	Server1	Server	2/13/2017 12:04:48 AM	2/13/2017 9:33:51 AM
DefaultEntity.16	DefaultEntity.16	Server1	InputBuffer	Sproing	Input@Server1	Input@Server1	BasicNode	2/13/2017 12:30:15 AM	2/13/2017 1:00:15 AM
DefaultEntity.15	DefaultEntity.15	Server2	Processing	Zot	Output@Server2	Output@Server2	TransferNode	2/13/2017 12:30:30 AM	2/13/2017 1:00:30 AM

When the user executes one of these steps with *Notification Type* set to 'StartConstraint', we create a new entry in the Constraint Log, setting the *Start Time* column to TimeNow. Then later, when the user executes one of these steps with *Notification Type* set to 'EndConstraint', with matching values for the other properties, we "close" that entry in the log by setting the *End Time* column to the current TimeNow.

New Dynamic Selection Rule Property – Filter Expression

The Dynamic Selection Rule grouping of properties has been enhanced to include a *Filter Expression*. This allows a user to optionally filter out any candidate entities that don't satisfy a specified logical condition. It is like the *Filter Expression* property provided by the 'Smallest Value First' and 'Largest Value First' rules.

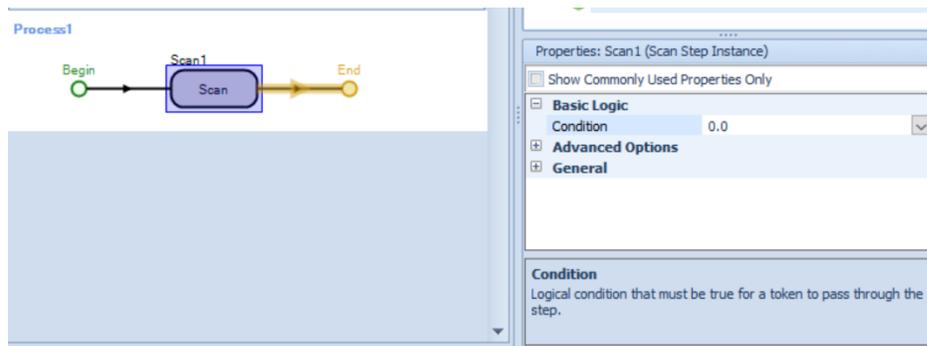


Simio Release 9 – Sprint 151 – February 7, 2017

In this sprint, we have added a new polled waiting approach for modeling with the Scan step. Additionally, we have enhanced our data binding options such that 'bound' data tables can be temporarily disabled for model distribution and testing.

New Scan Step

We have added a Scan step which provides a new polled waiting approach that allows a modeler to hold a process token until a specified condition is true, where the logical condition may be specified as any arbitrary expression.



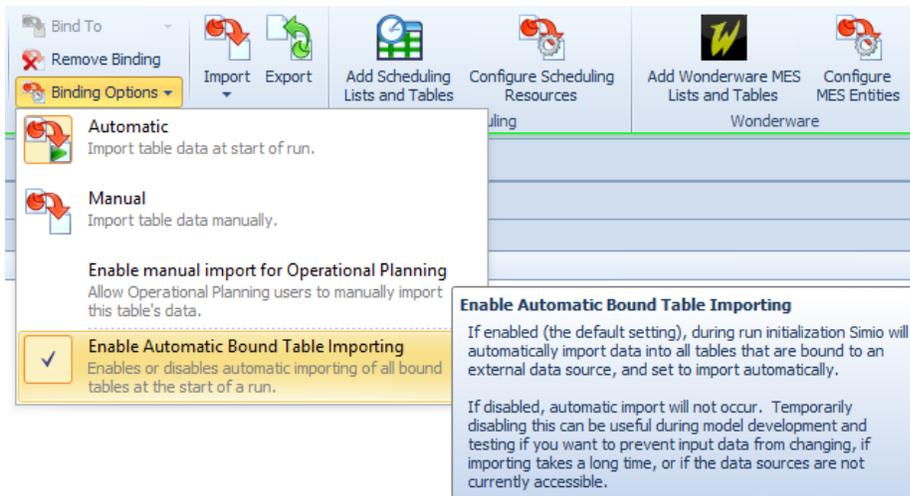
See more details in the Simio help for specifics of how this step works.

Enhancement to Resource Plan Gantt - Selecting a row(s) displays the property grid for the selection. (Enterprise Edition)

We have enhanced the Resource Plan Gantt so that when a user clicks on a resource name in the Resource Plan, the Resource related tables are displayed in the property view. This is useful, for example, if there is a column in the Resources table than allows the user to specify the work schedule. The enhancement allows the work schedule to be changed for a Resource from the Planning tab.

Option to Toggle Data Table Bindings

There is a new menu item on the Data tab under Binding Options that lets users to NOT do the automatic importing it would normally do for data bound tables. This is a model-level setting that is saved to the project file.



Note that this does not “temporarily *remove* table bindings” but will bypass the automatic import at start of run. When a model is in this state, bound tables that are set to automatically import show the following in red:

Table1	
Bound to Excel: C:\SimioModels\SAMPLE_DATA_FOR_BINDING_TO_TABLE.xlsx, Worksheet: Sheet1	
<i>Automatic importing of bound tables is currently disabled.</i>	
Name	Value

Simio Release 9 – Sprint 150 – January 24, 2017

In this sprint, we have made several user-requested features including enhancements related to task sequences and functions. The Snap to Grid feature has also been changed to allow easier placing of objects within the initial Facility window configuration.

Server / Combiner / Separator – Task Sequences Enhancement

If using the 'Task Sequence' for processing in Server, Combiner, or Separator, and a task's *Process Type* is specified as 'Submodel', the user now has an option to copy over the attributes of the original entity to the created submodel entity (e.g., state values and table references). Previously only the 'Create New (No Copying)' behavior was supported. This new feature allows for greater flexibility for submodel type processing with the task sequences.

Task Information	
Sequence Number	10
Name	
Branch Type	Always
Process Type	Submodel
Submodel Entity Creation Method	Create New (No Copying) ▼
Submodel Entity Type	Create New (No Copying)
Submodel Starting Node	Copy Original Entity Attributes
Save Original Entity Reference	
Auto Cancel Trigger	All Immediate Predecessors Cancelled

Submodel Entity Creation Method
The method used to create an entity that will be sent to some other distinct part of the overall model. The task will be considered finished once this created submodel entity has ended its processing (i.e., been destroyed).
If copying the original entity's attributes, then all possible state values and table references will be copied from the original entity to the created submodel entity.

New Token Functions

We have un-deprecated the `Token.ContextObject` function (as requested by a customer) and for completeness, have added `Token.TaskInfo.IDNumber` function.

Token.TaskInfo.IDNumber - Returns the unique integer identifier number of the task assigned to the token. It is valid when using task sequences and the Task Precedence Method of either 'Immediate Predecessors Method' or 'Immediate Successors Method' where the task *ID Number* is specified.

New String Function

We have added a new string function that will compare sequence number strings.

String.CompareSequenceNumbers(sequenceNumber1, sequenceNumber2) - Compares two specified sequence number strings and returns 0 if the two sequence numbers have no implied dependency relationship, -1 if an item assigned the first sequence number must come before an item assigned the second sequence number, or 1 if an item assigned the first sequence number must come after an item assigned the second sequence number.

Each sequence number argument may be expressed as a string that contains either an integer or a dot-delimited sequence of integers (e.g., 10.2.3.1). Or, in a data table, you may add a property (column) of type 'Sequence Number' and the arguments of this function may be references to row values in such table columns.

Examples:

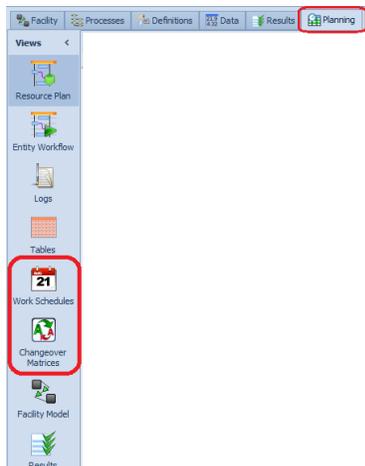
String.CompareSequenceNumbers("10", "10") will return 0.
String.CompareSequenceNumbers("10", "20") will return -1.
String.CompareSequenceNumbers("20", "10") will return 1.
String.CompareSequenceNumbers("10.1", "10.2") will return 0.
String.CompareSequenceNumbers("10.1", "20.1") will return -1.
String.CompareSequenceNumbers("20.1", "10.1") will return 1.

User Interface – Snap to Grid

The Snap to Grid has been updated slightly to snap to a quarter of a grid line in the default Facility window upon opening a model. This provides more flexibility in placing objects while maintaining the snap to grid feature 'on' setting at the start of the model building process.

Simio Enterprise – Planning Tab Enhancement

Within the Planning tab of Simio Enterprise Edition, we have added the Work Schedules and Changeover Matrices panel buttons to allow easy access to creating/modifying either resource work schedules or changeover matrices times. These are also available in 'Scheduler mode'.



Updated Visual Studio Templates

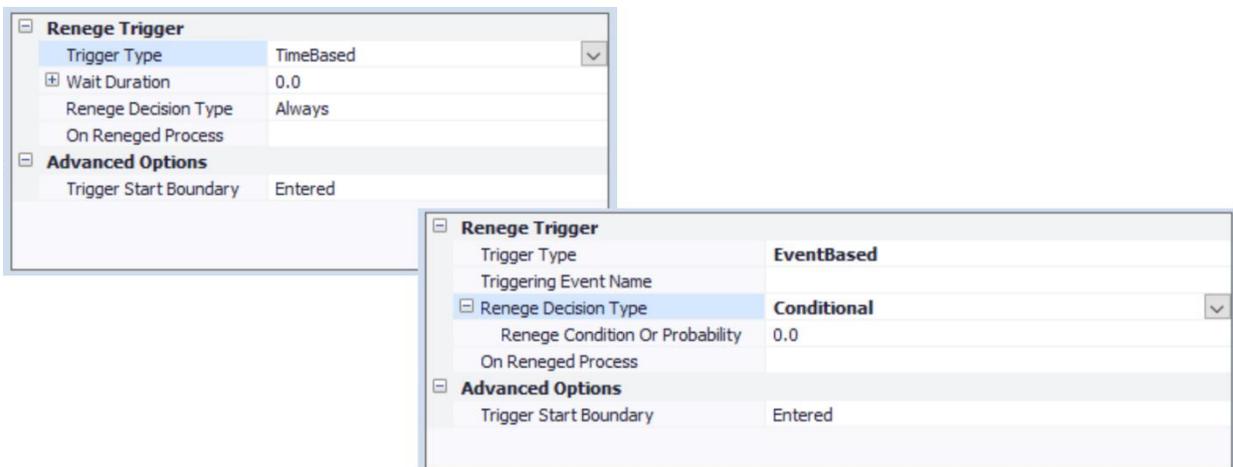
We have updated our C# Visual Studio Templates to support Visual Studio 2012, 2013 and 2015, and have added corresponding templates for VB.NET.

Simio Release 9 – Sprint 149 – December 13, 2016

In this sprint, we have enhanced the buffer logic on Standard and Flow Library objects to include renegeing, or abandoning waiting in a queue or station. Models illustrating various types of balking and renegeing have also been added to our extensive SimBit library.

Station Element - Renegeing

The Station element has been enhanced to include renegeing, when an entity decides to abandon waiting in a queue or station. Within the Advanced Options properties, a Renege Triggers repeating property editor allows users to define multiple triggers for renegeing. Triggers can be Time or Event based, and once the trigger has occurred, the renegeing can happen always or based on a probability or condition of the system.



There is also a new function, **NumberReneged**, that returns the total number of entities that have abandoned waiting in the station.

Buffer Logic – Renegeing Options

In sprint 146, we added balking options to the input and/or output buffers many objects. We have continued that effort and now provide renegeing options for the buffer(s) within the Source, Server, Combiner, Separator and Workstation within the Standard Library, as well as the Filler, Emptyier, ItemToFlowConverter and FlowToItemConverter within the Flow Library.

For each input buffer and output buffer, the Balking & Renegeing Options includes the balking properties as well as a Renege Triggers repeating property editor to specify one or more methods to trigger renegeing within a buffer.

Within the Reneges Triggers repeating property editor, there is a *Trigger Type* property to allow for either ‘Time Based’ or ‘Event Based’ triggering. For Time Based triggering, a *Wait Duration* expression is specified, while for Event Based triggering, the *Triggering Event Name* property is displayed.

If reneging is triggered, the action can happen either always (no conditions), or based on a probability or condition of the system. This allows for logic such as an entity evaluating the buffer size after a wait duration to determine whether to remain in the queue. The reneged entity can either be destroyed or sent to a different node for alternative processing.

See the associated SimBits or the Simio Help page for Balking and Reneging Options for more examples.

Trigger Type
The type of trigger.

Wait Duration
The wait duration until a renege decision by an entity waiting in the buffer.

Renege Decision Type
The method used by an entity when the trigger occurs to decide whether to renege from the buffer.

Renege Node Name
Facility node location to send an entity that has renege from the buffer.

Triggering Event Name
The name of the event whose occurrence will trigger a renege decision by an entity waiting in the buffer.

Renege Condition Or Probability
The renege condition or probability specified as an expression. If a probability then enter the chance of renege as a value between 0.0 (0%) and 1.0 (100%).

Trigger Start Boundary
Indicates when the trigger becomes active for an entity occupying the buffer. Can be either immediately when the entity has entered the buffer or, alternatively, not until its transfer into the buffer has been ended (e.g., after a transfer-in time or any other entry related logic).

Simio also includes statistics on the total number of renege entities for a buffer in the automatically reported results.

Drop Filter Fields Here

Average

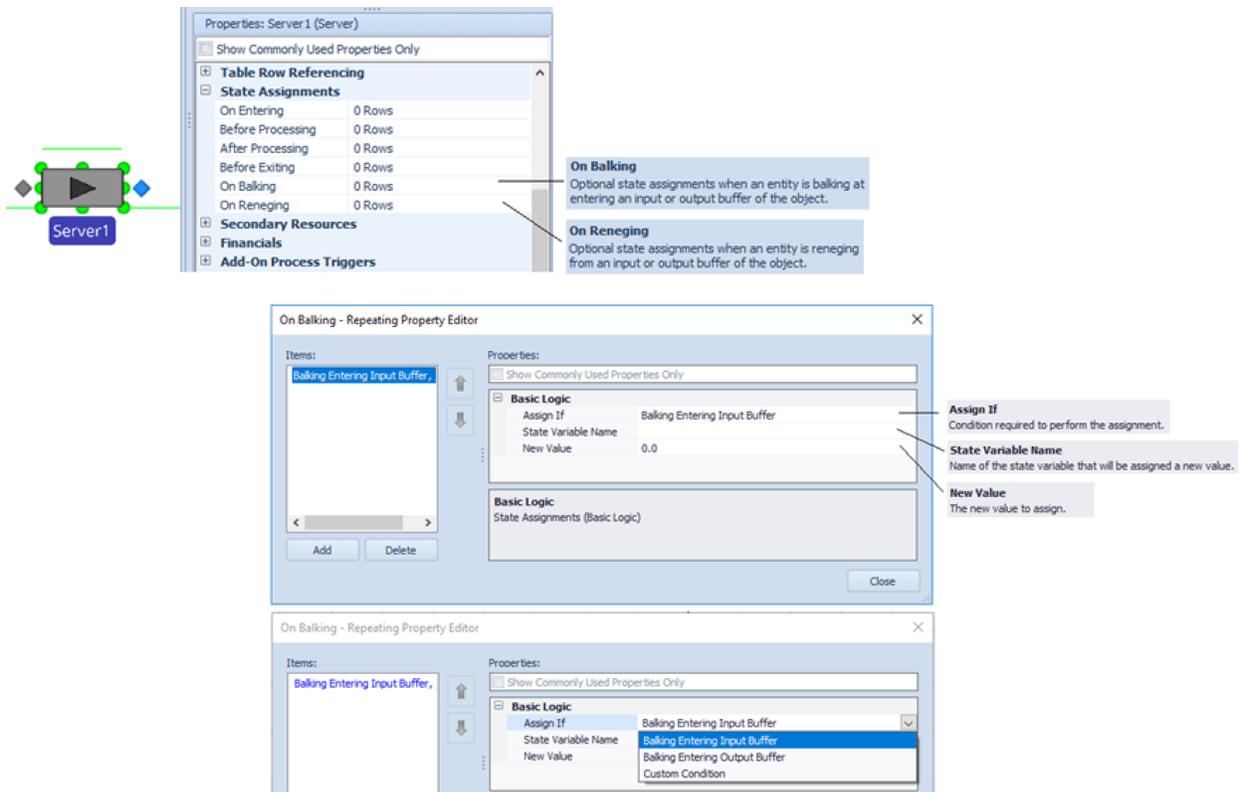
Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
Server	Server 1	InputBuffer	Content	NumberInStation	Average	0.1169
					Maximum	4.0000
			HoldingTime	TimeInStation	Average (Hours)	0.0010
					Maximum (Hours)	0.0132
					Minimum (Hours)	0.0000
			Throughput	NumberBalked	Total	2,843.0000
				NumberEntered	Total	2,847.0000
NumberExited	Total	2,847.0000				
	NumberReneged	Total	12.0000			

Drop Column Fields Here

Server1.InputBuffer.NumberReneged

State Assignments - On Balking and On Renegeing

We have enhanced the Properties window of selected Standard or Flow Library objects to allow optional state assignments whenever an entity is balking at entering a buffer or renegeing from a buffer. Options are available for making assignments under different conditions.



Failed and Repaired Events

The 'Failed' & 'Repaired' events for Standard and Flow Library objects that have resource capabilities and failures are now accessible. These events may now be used in Renege Triggers as event based triggers, thus minimizing or eliminating any excess process logic. For example, the SimBit ChangingQueuesWhenServerFails now accesses the Server object Failed / Repaired events to trigger reneging to the alternative Server, thus eliminating the Decide/Search/Remove/Assign step approach previously used.

New SimBits – Balking and Reneging

We've added 3 new SimBits to demonstrate the Balking and Reneging functionality. Additionally, the existing SimBit ChangingQueuesWhenServerFails.spfx was changed to reflect the new balking and reneging logic of the Standard Library Objects.

SourceWithBalkingIfBlocked.spfx – This model shows the balking of entity arrivals at a source if there is no immediate space at the downstream server.

ServerQueueWithBalkingAndReneging.spfx – This model includes a single server queue with balking and reneging (impatient customers). Customers will balk from the queue if it's above a certain length. Then, once customers enter the queue, they have a waiting time after which they evaluate their place in the line to determine whether or not to renege.

MultiServerSystemWithJockeying.spfx – This model is service system that consists of a group of parallel servers, where each server has its own waiting line. Customers switch between lines if they think they will get served faster.

Entity 'Queueing' Functions

To make it easier for a modeler to check in a conditional expression whether an entity is currently waiting for a specific type of constraint, the following new functions will be provided for an entity object:

Queueing.IsWaitingResourceAllocationQueue - Returns True (1) if the entity is currently waiting in the AllocationQueue of a resource. Otherwise, the value False (0) is returned.

Queueing.IsWaitingMaterialAllocationQueue - Returns True (1) if the entity is currently waiting in the AllocationQueue of a material. Otherwise, the value False (0) is returned.

Queueing.IsWaitingStationEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a station. Otherwise, the value False (0) is returned.

Queueing.IsWaitingLinkEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a link. Otherwise, the value False (0) is returned.

Queueing.IsWaitingNodeEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a node. Otherwise, the value False (0) is returned.

Queueing.IsWaitingRidePickupQueue - Returns True (1) if the entity is currently waiting in the RidePickupQueue of a node to be picked up by a transporter. Otherwise, the value False (0) is returned.

Queueing.IsWaitingRouteRequestQueue - Returns True (1) if the entity is currently waiting in the RouteRequestQueue of a routing group element to be assigned a destination. Otherwise, the value False (0) is returned.

Queueing.IsWaitingBatchLogicParentQueue - Returns True (1) if the entity is currently waiting in the ParentQueue of a batch logic element to collect a batch of other entities. Otherwise, the value False (0) is returned.

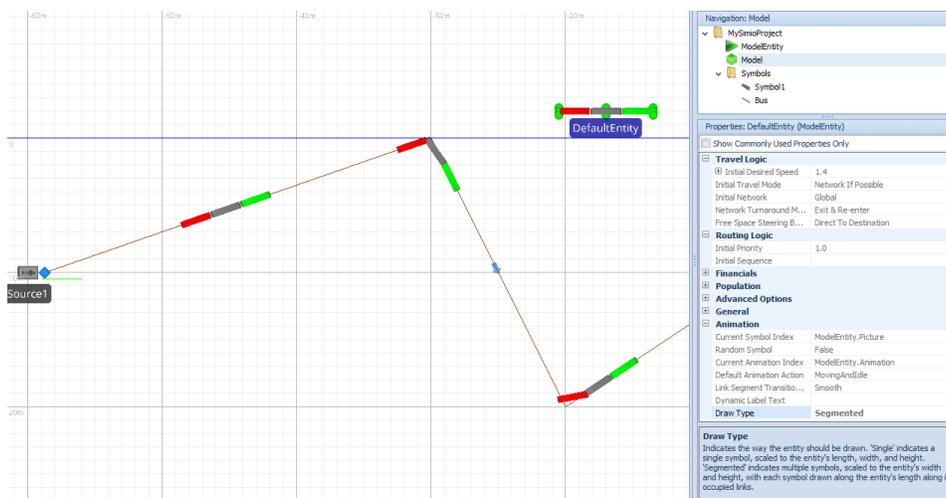
Queueing.IsWaitingBatchLogicMemberQueue - Returns True (1) if the entity is currently waiting in the MemberQueue of a batch logic element to be added as a member to a batch. Otherwise, the value False (0) is returned.

Simio Release 9 – Sprint 147 – October 28, 2016

In this sprint, we have added several new features, from new state assignments and table referencing within our Server type objects to segmented entity movement for smoother animations of certain symbols. Our Support ribbon options for Videos and Training have also been updated to provide users quick access to many resources for learning Simio and viewing examples.

Entity Animation - Display entities as segmented along links

Entities now have a *Draw Type* property (Animation) that allows the entity to be 'Single' or 'Segmented'. To have the entity actually move segmented, both the new *Draw Type* property should be set at 'Segmented' and the entity symbol itself must be able to be segmented. For example, to animate a train, the user would go to the Project Home ribbon, click New Symbol -> Create New Symbol, and place a number of individual cars and engines into the symbol. Then the new symbol can be applied to the ModelEntity placed in the Facility window.



State Assignments in Library Objects

The State Assignments section of properties in the Server, Combiner, Separator, Filler and Emptier objects has been enhanced to include a repeating property editor for optionally making assignments *Before Processing* and/or *After Processing* at the object.

Properties: Server1 (Server)

Show Commonly Used Properties Only

Process Logic

Capacity Type: Fixed
 Initial Capacity: 1
 Ranking Rule: First In First Out
 Dynamic Selection Rule: None
 Transfer-In Time: 0.0
 Process Type: Specific Time
 Processing Time: Random.Triangular(.1,.2,.3)
 Off Shift Rule: Suspend Processing

Buffer Logic

Reliability Logic

Table Row Referencing

State Assignments

On Entering: 0 Rows
 Before Processing: 0 Rows
 After Processing: 0 Rows
 Before Exiting: 0 Rows

Secondary Resources

Financials

Add-On Process Triggers

Advanced Options

General

Animation

Before Processing
 Optional state assignments when an entity has been allocated capacity to be processed at the object, but before entering (or ending transfer into) the object's processing station.

After Processing
 Optional state assignments when an entity has completed its processing and is about to attempt its exit from the object's processing station.

Table Row Referencing in Library Objects

The Table Reference Assignments section of properties in the Source has been renamed to Table Row Referencing. Additionally, an *Action Type* property has been added to allow users not only to 'Reference Existing Row' in a table before or on creating entities, but also to allow for alternatively to 'Add New Row' to an output table.

Properties: Source1 (Source)

Show Commonly Used Properties Only

Entity Arrival Logic

Entity Type: DefaultEntity
 Arrival Mode: Interarrival Time
 Time Offset: 0.0
 Interarrival Time: Random.Exponential(.25)
 Entities Per Arrival: 1

Stopping Conditions

Buffer Logic

Table Row Referencing

Before Creating Entities

Action Type: Reference Existing Row
 Table Name:
 Row Number:

On Created Entity

State Assignments

Financials

Add-On Process Triggers

Advanced Options

General

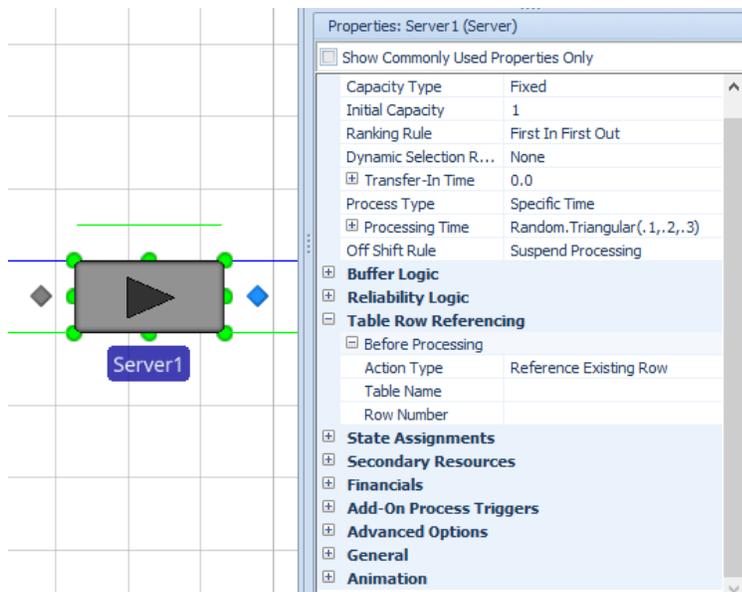
Animation

Action Type
 The type of table row reference action to perform.

The 'Reference Existing Row' action type may be used to set a reference to an existing row in a data table or sequence table. The row index into the table is specified by the Row Number property.

The 'Add New Row' action type may be used to add a new row to an output table. A reference will be automatically set to the newly added row.

In conjunction with this change, the Server-oriented objects including the Server, Combiner, Separator, Filler and Emptier objects, have been enhanced to include the ability to reference table rows before the entity starts processing (prior to before processing state assignments). Each of these objects now allow for either referencing an existing row in a data or sequence table, or adding a new row to an output table.

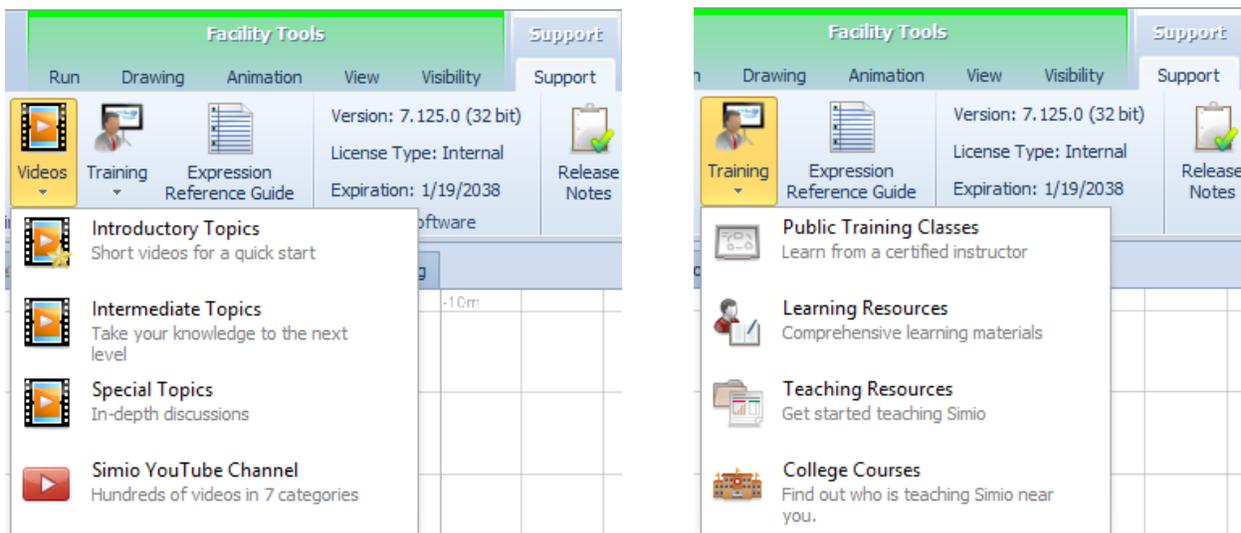


SetRow and AddRow Step Enhancements

The SetRow step (available in all Simio editions) and the AddRow step (available in Simio Enterprise edition) have both been enhanced to include a new conditional expression property. If a condition is specified, then it must evaluate to 'True' to perform the set row or add row action.

Support Ribbon Updates

We have updated both the Videos and Training buttons on the Support ribbon to provide a more organized and up to date look at the various levels for learning more about Simio, as well as resources for training. All options take you to links on our Simio website.



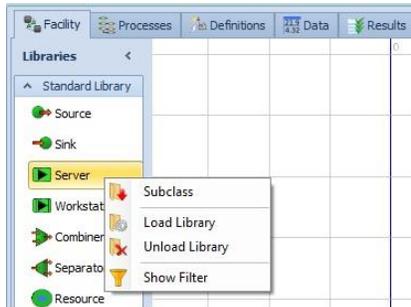
Remove Step Enhancements

We have changed the Remove step to include the ability to remove an entity from material allocation queues, as well as from network visit request queues (Global.VisitRequestQueue, for example). Entities either waiting for material or transporter (Worker/Vehicle) allocation can now be easily removed from

those queues if desired. This additional functionality has been added as we continue to enhance our balking / reneging capabilities.

Right Click on Library Panel Change

In addition to using the Load Library button from the Project Home ribbon to load a library into the project, users can now also right click on the library area itself to access the Load Library functionality. This right click area now allows both loading and unloading of library files.



Performance Enhancements with Status Label Creation

We've changed the way status labels are created - specifically, this addresses some performance issues with many status labels attached to many, many dynamic objects (entities). The enhancements are most notable when many status labels have the same exact size, color, and content, as we now share the same generated label between all those instead of creating a new one for each of them.

New Functions

LocationAt.Geographic - converts a latitude and longitude to an X, Y, Z location in the model, relative to the current latitude and longitude of the model's origin. The result of the function is a Location, with Z and X components that correspond to the latitude and longitude, respectively. Note that the range of valid values for latitude is -90.0 to +90.0, and for longitude is -180.0 to +180.0. If either value is out of range, the result is a Location consisting of all NaN values.

DateTime.SystemNow - returns the current DateTime from the computer (actual time, not simulation time), expressed in the computer's current timezone.

DateTime.SystemNowUtc - returns the current DateTime from the computer (actual time, not simulation time), expressed as Coordinated Universal Time (UTC).

Important Notice – Windows Vista

Please note that this sprint Simio 9.147 is the last release that we'll support running on Windows Vista, since Microsoft's extended support for Vista ends in April 2017.

Where's the Rest?

In case you are a history buff who would like to see a sprint by sprint account of how we got to this point, you can find that here:

<http://www.simio.com/downloads/public/documents/SimioHistoricalReleaseNotes.pdf>. This contains records from the first 146 sprints (0-146) covering the generation of Simio major releases 1 - 8.