Simio Historical Release Notes

Activities leading up and including Sprints 0-260

General Information

These notes are for all our interim releases (Sprints) from inception up to Sprint 260. If you are looking for something more current, please go to:
http://www.simio.com/download/

Simio Release 16 – Sprint 260 – February 29, 2024

Welcome to Sprint 260! In this Sprint, you will notice a new property on the Counter element as a new model in the Network Shortest Paths SimBit.

Desktop Edition Features:
Counter Element Enhancement – This feature adds a *Value Expression* property to the Counter element which allows you to specify a value to increment/decrement the count by.

New SimBit Model – Tiebreaker Fewest Turns – This SimBit demonstrates a Network using fewest turns when there are ties for shortest routes.

Simio Release 16 – Sprint 259 – February 6, 2024

Welcome to Sprint 259! In this Sprint, you will notice a few animation enhancements as well as a new About Simio button. Simio Portal Edition received a data table enhancement.

Desktop Edition Features:
Ability to Add Animation to Node Definitions – This user requested feature allows you to add labels, queues, and animation to custom Node definitions.

Additional Symbol Animation Support – This feature allows symbols with node animation can now be loaded into Simio.

About Simio Enhancement – This feature adds an About Simio button to the Support ribbon which will show your Simio Version, License Type, Registered to, Expiration, and .NET Framework version. It also provides a link to the License Terms as well as a direct link to email your important information into Simio.

Portal Edition Features:
Data Table Enhancement – This feature keeps filters on data tables when editing, adding or removing a row in order to easily edit tables.

Simio Release 16 – Sprint 258 – January 26, 2024

Welcome to Sprint 258! In this Sprint, you will notice an enhancement to the Expression Editor as well as automatic bound table imports. Creating objects from a data table also received an animation enhancement. Simio Portal Edition now has a Download Support Files page.

Desktop Edition Features:
Expression Editor Enhancement – This user requested feature turns off the filtering of the Expression Editor by default. This was implemented to allow you to view more expression segments easier.

Automatic Bound Table Import Enhancement – This feature enables the Insert Row and Remove Row buttons when a table is automatically bound to an external data source if the Enable Automatic Bound Table Importing toggle is off.

Auto-created Object Enhancement – This user requested feature allows you to set the auto-created object's display symbol via the *Symbol Index* property of the Object Property column of a data table.

Portal Edition Features:
Download Files for Support – This feature allows a Portal Admin to download the Web Config, SAML Log, App Data, Audit Log, and APP Config for a certain time period in order to help debug Simio Portal Edition.

Simio Release 16 – Sprint 257 – December 8, 2023

Welcome to Sprint 257! In this Sprint, you will notice three Log enhancements, a new element, and a new reference property.

Desktop Edition Features:
Generic Element Reference Property – This user requested feature lets User Defined Elements to now have corresponding Element Reference Properties available in the Definitions tab. Simio's default User Defined Elements will appear in the bottom section of the Element Reference Property drop-down. Any additional custom User Defined Elements will also appear automatically.

Tally Observation Log Enhancement – This feature allows Expression columns in the Tally Observation Log to include expressions referencing table rows referenced by the Tally Statistic element and expressions referencing the Tally step token's associated object or context object, or table rows referenced by that token's associated object or context object or the token itself.

Counter Element – This new element may be used to track attribute-based counts of entities in the system. The Counter can count either the 'NumberWaiting' or the 'NumberEnroute'.

Transportation Log and Constraint Log Enhancement – This feature lets Expression columns added to these two logs to have the property *Show In Gantt Tooltips*. When set to 'True', it will display the log expression from the Transport Log or Constraint Log on the corresponding Gantt bar.

Simio Release 16 – Sprint 256 – November 7, 2023

Welcome to Sprint 256! In this Sprint, we added the Path Planner Element. A histogram was added for data columns. We wrapped up this Sprint with an Interrupt Step enhancement and an enhancement to the Entity States Spreadsheet.

Desktop Edition Features:
Ability to Watch an Entity from Entity States Spreadsheet – This user requested feature allows you to Watch an entity directly from the Entity States Spreadsheet. Right clicking on the desired entity in the Entity States Spreadsheet and selecting Watch will allow you to easily watch that entity without having to navigate in the Facility Window.

Path Planner Element – This new Element may be used in conjunction with a Transfer step to find and reserve a deadlock-free shortest network path for an entity from its current node to a destination node.

Interrupt Step can Check Tokens – This user requested feature allows the Interrupt step to have a better understanding of information on the Token so that you can better control interrupting a Token.

Data Column Histogram – This user requested feature allows you to create a histogram from a specified column of a data table.

Simio Release 16 – Sprint 255 – October 3, 2023

Welcome to our Release of Simio 16! In this Sprint, you will notice a new inventory control and replenishment methodology. Numerous Network element enhancements were made, and a new unit of measurement was added. Lastly, there are three new SimBits and one new Example.

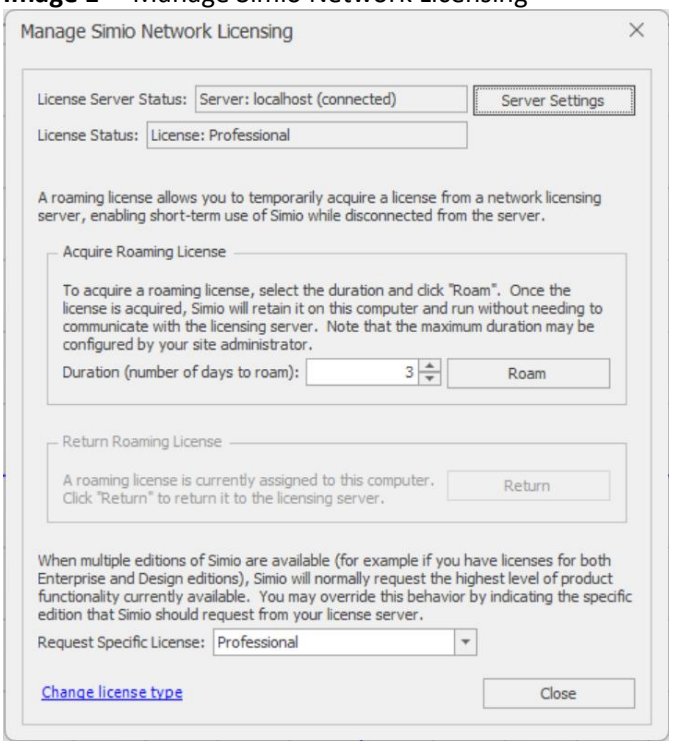
Attention! Users of Simio's on-premises network license type, managed by Reprise License Manager ("RLM"), must have an updated license server to use Simio 15.254 and newer. Simio 15.254 and newer should not be installed on client machines before RLM is updated to v15.1. Reprise license keys are in the format of four groups of four digits, e.g., xxxx-xxxx-xxxx-xxxx. Clients can verify their license type by visiting File > Licensing in Simio, where users of Reprise on-premises network licenses will show a dialog like in Image 1. If you are unsure if this requirement pertains to you, please verify your license type with your administrator or organization's IT/infrastructure team. Users of on-premises network licenses provided/managed by QLM are not affected.

In usual configurations, the license server is installed on a central machine, managed by the organization's IT/infrastructure team, and is not installed on client/end user machines.

RLM 15.1 can be found at https://www.simio.com/downloads/public/software/RLMLicenseServer.php

Please pay close attention to "02 README Updating from Previous Version" contained therein.

Image 1 Manage Simio Network Licensing



Desktop Edition Features:

Demand Driven Material Requirements Planning (DDMRP) – RPS, Partner, and Academic Editions Only – This feature also known as "DDMRP", revolves around strategically placed stock positions called buffers that decouple demand from supply. These inventory buffers are comprised of three color-coded zones; Green, Yellow, and Red. Simio now includes four DDMRP calculators: Average Daily Usage, Decoupled Lead Times, Buffer Zone Sizes, and Qualified Spike Demand.

Network Element Enhancements – Shortest Paths Enhancements – These enhancements add four new properties to the Network element. *Shortest Paths Graph Type*, *Shortest Paths Exclusion Condition*, *Shortest Paths Weight Multiplier*, and *Shortest Paths Tie Breaker Rule* were added to allow users to enhance their modelling to include more complex Networks makeups.

Oil Barrel Added as an Option for Volume and Flow Rate – This user requested feature allows you to select oil barrels as a unit of measurement in Simio.

New SimBit – Business Graphics Animation – This SimBit highlights and teaches the enhanced animation features added to Simio over the last few Sprints.

New SimBit – Efficient Search Step Practices – This SimBit shows numerous ways to use a Search step efficiently.

New SimBit – Network Element Shortest Paths Logic – This SimBit highlights and teaches the new Network element shortest paths features added to this Sprint.

New Example – DDMRP Example – This example uses Simio RPS Edition to build a data-generated model for orders, within multiple manufacturing facilities and a distribution center, using Demand-Driven Material Requirements Planning (DDMRP) methodologies.

Simio Release 15 – Sprint 254 – July 25, 2023

Welcome to Sprint 254! In this Sprint, you will notice enhancements to the Material Usage Log as well as the Network Element. Additionally, Data Tables can now be categorized.

Note: Simio Personal Edition has been deprecated from the software. This Sprint introduces Simio Trial Edition. Simio Trial Edition is a no cost version of Simio that expires thirty (30) days after installation and has no limits on what can be modeled. It is perfect for exploring and learning Simio's simulation capabilities.

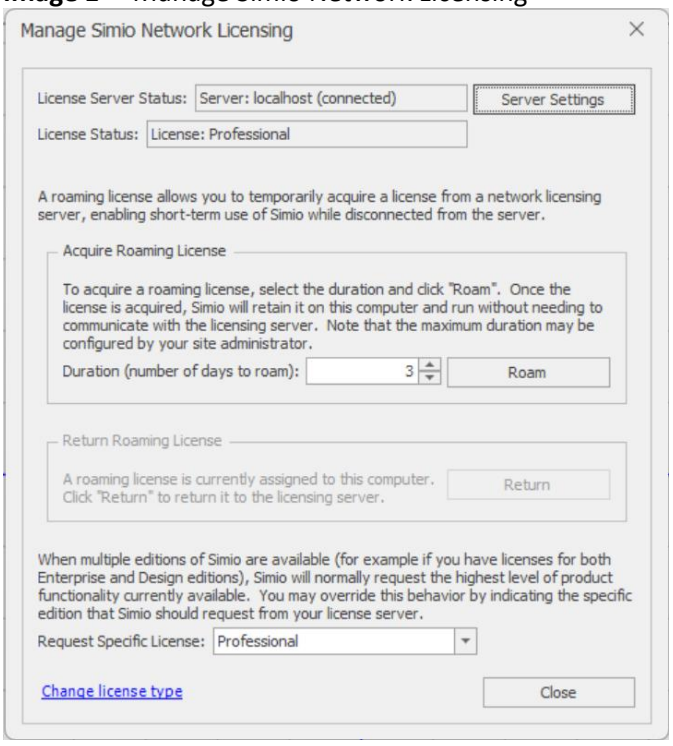
Attention! Users of Simio's on-premises network license type, managed by Reprise License Manager ("RLM"), must have an updated license server to use Simio 15.254 and newer. Simio 15.254 and newer should not be installed on client machines before RLM is updated to v15.1. Reprise license keys are in the format of four groups of four digits, e.g., xxxx-xxxx-xxxx-xxxx. Clients can verify their license type by visiting File > Licensing in Simio, where users of Reprise on-premises network licenses will show a dialog like in Image 1. If you are unsure if this requirement pertains to you, please verify your license type with your administrator or organization's IT/infrastructure team. Users of on-premises network licenses provided/managed by QLM are not affected.

In usual configurations, the license server is installed on a central machine, managed by the organization's IT/infrastructure team, and is not installed on client/end user machines.

RLM 15.1 can be found at https://www.simio.com/downloads/public/software/RLMLicenseServer.php

Please pay close attention to "02 README Updating from Previous Version" contained therein.

Image 1 Manage Simio Network Licensing



Desktop Edition Features:

Material Usage Log - Expression Evaluation Enhancements – This user requested feature allows custom expression columns to:

1. Get an attribute value of the object (entity) that has consumed or produced the material using the *Owner* keyword. For example, *Owner.Entity.Priority*.
2. Get a table row value referenced by the object (entity) that has consumed or produced the material using the *Owner* keyword. For example, *Owner.TableName.ColumnName*.
3. Get an attribute value of the site object if the material is location-based inventory. For example, *Site.StateName* if the site is an instance of an object class named 'Site'.
4. Get a table row value referenced by the site object if the material is location-based inventory. For example, *Sites.ColumnName* where 'Sites' is a data table defining all sites in the model.
5. Get a table row value referenced by the material. For example, *Materials.ColumnName* where 'Materials' is a data table defining all materials in the model.
6. Get a table row value referenced by the token that executed the Consume or Produce step in the process logic. For example*, ProcessingTasks.ColumnName* where 'ProcessingTasks' is a table defining the processing tasks for a task sequence at a Server.

Data Table Categorization – This user requested feature allows you to assign a *Category* to Data Tables. When you choose a Selected Table Category from the dropdown, Simio will only show the Data Tables that correspond to that *Category*.

Network Element Enhancements – These user requested features were implemented to improve usability of the Network Element. First, Network membership can now be set via Data Tables. Second, the Global Network is now exposed. Finally, the *Links Type* property can be set to 'All' to capture all links in the system.

Simio Release 15 – Sprint 253 – June 20, 2023

Welcome to Sprint 253! In this Sprint, you will notice an additional Log Export Option as well as a new property pertaining to Task Sequences. Lastly, you will be able to change Axis Scale of Dashboards. Simio Portal Edition now provides the ability to reassign model ownership.

Desktop Edition Features:
Log Export Option – Export At End Of Experiment Replication Run – This user requested feature allows you to export the specified Log for each Scenario and Replication in an Experiment when toggled on.

Invalid Task Dependency Notification Type – This user requested feature is a new property on the Task Sequence Element, Combiner, Separator, and Server objects that indicates the level of alert at runtime if the task sequence has missing, duplicate, or otherwise invalid data defining the task dependency relationships. There are three options: Error – Throw runtime error, Warning – Throw runtime warning and skip the invalid data, Trace – Write to trace and skip the invalid data.

Ability to Change Axis Scale of Dashboard – This user request allows you to specify axis scales, maximums and minimums, and ticks. This was designed to provide maximum flexibility for Dashboard display.

Portal Edition Features:
Ability to Reassign Ownership of Model – This feature allows an Admin or the model owner to reassign ownership of the model to another user within your Portal.

Simio Release 15 – Sprint 252 – April 28, 2023

Welcome to Sprint 252! In this Sprint, you will notice enhancements to the Expression Editor, table importation, and the Documentation Report. A new SimBit was also added during this Sprint. Portal gained the ability to view Sankey Diagrams and the Published Plan view now shows additional information.

Note: Though never guaranteed, Simio has made an effort to ensure models created in newer versions could be opened/viewed in older versions. They likely could not run, but they could be reviewed. In release 15.252, we made a change to the way Simio saves project files to help organizations compare Simio projects at a file level. Any model saved in 15.252 or newer will error when opening in version 15.250 and older. Moving forward, Simio's intent is to again allow models created in newer versions to be opened in 15.252 and above.

Desktop Edition Features:
Documentation Report Enhancements – This feature allows for table schema descriptions to appear in the Documentation Report

Expression Editor Enhancement – This user requested allows you to space out an expression in the Expression Editor. Using Ctrl + Enter will create a new line and using Ctrl + Tab will create a tab. Both of these tools can be used to better organize an expression.

New SimBit – Experiment With Data Configurations – This SimBit shows how to create multiple Bindings for a table or Configurations for a Data Connector while using an Experiment.

Table Importation Enhancement – This feature allows you to manually edit tables if *Enable Automatic Bound Table Importing* is disabled. Prior to this Sprint, you could only manually edit tables if table data was imported manually.

Portal Edition Features:
Support for New Dashboards – This feature allows for Sankey Diagrams to be viewed on Simio Portal Edition.

Additional Information Shown on Published Plan – This feature shows the Start Time, Ending Type, property values, and who published the Plan on the Published Plan view.

Simio Release 15 – Sprint 251 – March 28, 2023

Welcome to Sprint 251! In this Sprint, you will notice source control enhancements as well as an animation enhancement for custom links. We also removed an obsolete file from the installation.

Desktop Edition Features:
Ability to Set the Path Decorator on the Object Definition – This user requested feature allows Path Decorators to stay applied on a custom link.

Saving as Simproj Enhancement – Processes get Saved to their own XML File – This user requested feature was designed to support two modelers working on the same process. When *Save Project as Multiple Files* is set to 'True' and the file format is .simproj, Simio will produce a separate model file for Add-On Processes.

Source Control Enhancement – *Support Source Control* Property – This user requested feature changes a former property *Sign Files* to *Support Source Control*. Along with the name and description change, this property now will streamline the process of using source control by reducing the amount of information written out in the save history, if applicable.

Removal of Obsolete File – We have removed the CSVGridDataProvider example from the install.

Simio Release 15 – Sprint 250 – March 7, 2023

Welcome to Sprint 250! In this Sprint, you will see two new properties for Bar Charts, Pie Charts, and Time Plots when using a *Data Source* of an Output Table. You will also see enhancements made to our filtering as well as the State Statistic element. We also made a change to our supported *Graphics Types*.

Desktop Edition Features:
Bar Chart/Pie Chart/Time Plot *Summary Function* – This feature allows you to summarize data in one of Simio's charts when using a *Data Source* of an Output Table.

Bar Chart/Pie Chart/Time Plot *Aggregation Interval* – This feature allows you to aggregate data over a specified time frame when using a *Data Source* of an Output Table for Simio's charts.

Table, Log, and Trace Filter Enhancement – This user requested feature places an expression view at the bottom of each of the above filter locations which will allow you to create and edit filters more easily.

State Statistic Enhancement – This user requested feature allows the State Statistic element to Log Observations for non-numeric state values.

Graphics Type Change – OpenGL is no longer available as a *Graphics Type* in Simio.

Simio Release 15 – Sprint 249 – January 27, 2023

Welcome to Sprint 249! In this Sprint, you will notice significant animation changes, importer and exporter enhancements, a new function, and periodic views of Logs. RPS Users will have a new Advanced Option for ending a Run. We wrapped this Sprint up with a Portal Experimenter enhancement and a Pivot Grid enhancement.

Desktop Edition Features:
New Function – This feature adds the TableName.NumericPropertyName.Sum function to return the sum of the numeric property column's values in a specified range of rows based on an optional true or false condition.

Database Exporter Enhancements – This user requested feature allows commas to be used as a decimal separator.

Web API Importer Enhancements – This user requested feature adds two new properties to the Web API Importer: NumberOfRecordsParameter and StartingOffsetParameter.

Advanced Option – Stop At End Of Run – RPS Edition Only – This feature when toggled on will have the model run automatically stop without any user interaction. Prior to this feature, users would have to manually click the Stop button in order to stop the model.

Periodic Views of Logs – This feature provides additional Dashboard data sources that allows user to view various logs by day, hour, or week. You will be able to dashboard your results in accurate time periods.

Bar Charts – This user requested feature allows you to use Bar Charts in the Facility View of Simio. The Bar Chart can be set up to display data from Expressions, List States, Data Tables, or Output Tables.

Enhanced Animation – Table Based Time Plots, Bar Charts, and Pie Charts – This feature allows users to hook up any of the enhanced Charts to a Data Table or Output Table. Using a *Data Source* of either an Output Table or Data Table allows you to view the animation the same whether you Run the model or Fast-Forward the model.

Ability to Show Display Name in Pivot Grid – This user requested feature allows you to add an additional column to the Pivot Grid that will show the object's Display Name.

API Change - When running the SimioEngine using the API in a standalone or "headless" mode, the controlling program must now add references to Sqlite, since it is now incorporated into the SimioEngine. A recommended way is to use NuGet. For an example of this, see the GitHub project SimioLLC/RunSimioSchedule. The Sqllite reference is required for all Simio versions after 15.245.

Portal Edition Features:
Ability to Define Which Response is Primary Response in Portal Experimenter – This feature allows Experimenters to specify which of their Experiment's Responses is the Primary Response. This feature will enhance Experimenters ability to design their Experiments on Portal.

Simio Release 15 – Sprint 248 – November 22, 2022

Welcome to Sprint 248! In this Sprint, we improved our Status Pie animation and added tooltips to Portal Edition's Box and Whisker Plots.

Desktop Edition Features:
Status Pie Enhancements – This user requested feature provides animation enhancements to the Status Pie. You will now be able to specify the Status Pie as a "Pie", "Donut", or "Nested Donut" as well as setting custom *Color* expressions for the *Argument*.

Portal Edition Features:
Box and Whisker Plots Enhancement – This user requested feature shows tooltip information such as mean, confidence intervals, half width, etc. on BoxWhisker plots in order to provide more information for each Experiment replications for each Scenario. Hovering over the graphic will display this information.

Simio Release 15 – Sprint 247 – November 1, 2022

Welcome to Sprint 247! In this Sprint, we added a new replenishment policy and Log. We wrapped up this Sprint with an enhancement to the Watch Window and a VR enhancement.

Desktop Edition Features:
New Inventory/Material Replenishment Policy – You will notice a new *Replenishment Policy* of 'Demand-Driven MRP' on the Inventory and Material elements. DDMRP revolves around strategically placed stock positions called buffers that decouple demand from supply. These inventory buffers are comprised of three color-coded zones: Green, Yellow, and Red.

Inventory Review Log – **RPS Only** – This new Log automatically records inventory review events when the *Log Inventory Reviews* Boolean is togged on for the Material or Inventory element. This Log can be useful for determining inventory quantity, qualified spike demand, net flow position, zone sizes, and recommended order quantity.

Watch Window Enhancement – This user requested feature allows you to view the current table references for any associated modelling objects in the Watch Window to improve the debugging table references process.

Virtual Reality Enhancements – This user requested feature allows Oculus users to use the 'A' button on a controller to move to the location in the middle of their vision and the 'B' button to return to the origin of the Facility view.

Simio Release 15 – Sprint 246 – October 11, 2022

Welcome to Sprint 246! In this Sprint, we added a Reset Variables button. Also, you will notice an enhancement to our Status Pie and Status Plot. We also made a variety of improvements to performance.

Desktop Edition Features:
Reset Variables Button – This feature was designed to provide a way to reset the model without running the initialization logic or importing any automatic bindings. By selecting the dropdown on the Reset button, you can choose Reset Variables which will reset state columns, output tables, model results, and logs.

Enhanced Animation Plots and Pies – This feature was implemented to make the data source specification between Status Pie and Status Plot consistent. You will now be able to select what type of *Data Source* (i.e., 'Expressions', Logs, Tables, etc.) to use. Once you select the desired *Data Source*, additional properties will appear to give you the most flexibility with your animation.

Simio Release 15 – Sprint 245 – September 20, 2022

Welcome to Sprint 245! In this Sprint, we closed several bugs and made a few refactors to improve application performance.

Simio Release 15 – Sprint 244 – August 30, 2022

Welcome to Sprint 244! In this Sprint, you will see a new Experiment property on both Desktop and Portal. You will notice an enhancement to a recently added feature and an accessibility enhancement. This Sprint wrapped up with the addition of the Entity States Window.

Desktop Edition Features:
Accessibility Enhancement – This user requested feature allows you to use the Esc key on the keyboard to deselect an object in the Facility Window.

Response *Description* Property in Experiments – This user requested feature allows for a *Description* field to be added to each Response in an Experiment in order to provide more context for each Response.

Option to Turn Off Auto-Merged File from Write Step – This user requested feature allows you to toggle on/off the auto-merged write file from Experiments. The *Auto Merge Write Step Files For Experiment* property on the File element is set to 'False' by default. See Sprint 241's Release Notes for more information on this feature.

Entity States Window – This user requested feature gives a window to view all active entities in the model by entity type, displaying properties and states of each. This window allows a double click on any entity listed in the window and navigate to where the entity is in animation.

Portal Edition Features:
Response *Description* Property Displayed in Experiments – This user requested feature displays the *Description* field added to each Response in an Experiment in order to provide more context for each Response.

Simio Release 15 – Sprint 243 – August 9, 2022

Welcome to Sprint 243! In this Sprint, you will have access to a new animation type and there is an enhancement to Data Connectors.

Desktop Edition Features:
New Animation – Status Table – This feature was implemented to have an easy way to display information from a data table in the Facility View. The Status Table works with Properties and States in a Data Table and will automatically size the structure. You can set the colors of different areas of the Status Table. The Status Table will show the first 50 rows of data.

Multi-Line Data Connector String Property – This feature allows you to streamline steps needed to transform and cleanse data before importing the data into the Simio data tables.

Simio Release 15 – Sprint 242 – July 26, 2022

Welcome to Sprint 242! In this Sprint, you will notice a property change in Desktop as well as a new Export Status option in Portal. This Sprint also included a new step and a new Dashboard. This Sprint wrapped up with two Web API Connector enhancements and an improvement to the Object Updates Available window.

Desktop Edition Features:
States Enhancement – *Keep Values Between Runs* – This property replaces the property *Keep Values On Run Restart* Boolean property. Now you will have three options on keeping values between runs.

- 'Never' – Never keep the Output Table or State Column's values from the previous run when starting a new run
- 'Always' – Always keep the Output Table or State Column's values from the previous run when starting a new run.
- 'OnRunRestart' – Keep the Output Table or State Column's values from the previous run if a run restart occurs using the RestartRun step.

UnSetRow Step – This new step was designed to remove table row references previously set for the executing token or a specified object or element. The UnSetRow step can assist in resolving row reference conflicts.

Sankey Diagram Dashboard – This user requested feature gives an additional Dashboard option to display results. A Sankey Diagram is a visualization that can be used to depict a flow from one set of values to another.

Web API Connector Enhancements – These features include two enhancements to the Web API Importer and Exporter. First, the Web API Exporter can export multi-row. Second, a new authorization type "MSAL" has been added. MSAL silently authenticates the user using username and password.

Ability to Select/Deselect all Objects at once in Update Object Dialog – This user requested feature was designed to easily update all objects at once when updates are available.

Portal Edition Features:
Status for Manual Exports – This user requested feature shows the progress of a manual export from the Scheduler Mode in Portal. Using the Export Status selection from the Export Table drop down will show the status of the export from Portal.

Simio Release 15 – Sprint 241 – June 28, 2022

Welcome to Sprint 241! In this Sprint, will notice two material handling enhancements, a combined file created by the Write step on an Experiments with multiple replications, and a routing enhancement.

Desktop Edition Features:
Consume Step Enhancement – *Immediately Try Consume* – This user requested feature was implemented to improve Material modeling. This enhancement allows you to choose to immediately consume a Material at a Process step level instead of at a model level.

Produce Step Enhancement – *Immediately Try Allocate* – This user requested feature was implemented to improve Material modeling. This enhancement allows you to choose to immediately allocate a Material at a Process step level instead of at a model level.

Combined Files Created by Write Step when Running an Experiment with Multiple Replications –When a Model contains Write Steps in its Process Logic, and an Experiment is run, the output files can be merged for improved data portability, provided in addition to a file per Scenario per Replication. This was a user requested feature.

Ability to Use Constraint Logic on TransferNode – This user requested feature allows TransferNodes with an *Entity Destination Type* of 'BySequence' or 'Specific' to have additional Constraint Logic on them in order to ensure constraint requirements are met prior to routing to the next destination.

Simio Release 15 – Sprint 239/240 – June 7, 2022

Welcome to our release of Simio 15! In this Sprint, you will notice a Real Time Clock in RPS and Partner Edition. You will also notice enhancements to our WebAPI Connectors. This Sprint wrapped up with a Gantt enhancement in Portal Scheduler Mode and two new UTC functions.

Desktop Edition Features:
Real Time Clock – RPS and Partner Edition Only – This feature which was designed to assist in orchestrating the real time execution of the system modelled and adds a Real Time Clock button in the Run Speed section of the Run ribbon. When this is toggled on, the simulation clock will advance at the same rate as a real clock if running the model interactively.

WebAPI Data Connector Enhancements – This feature was added to improve the quality of our WebAPI Importer and Exporters by setting the way external data is imported/exported in a standard and proven way by passing parameters and transferring relevant data.

Ability to Delete Object by Right-Clicking – This user requested feature was implemented to improve Simio's useability. Now, you can right-click on an object or multiple objects and select "Delete" in the drop-down menu to remove an object from the Facility Window.

Additional UTC Functions – This user requested feature provides two new functions that can be used:
Run.TimeNowUtc: Returns the current simulation time in hours, expressed as Coordinated Universal Time (UTC).
Run.TimeNowUtcOffset: Returns the offset in hours between the simulation clock's time zone and Coordinated Universal Time (UTC) for the current simulation time.

Portal Edition Features:
Show Filtered Out Items button to Gantt ribbon in Portal Scheduler Mode – This user requested feature allows you to right-click on the Gantt ribbon to 'Show Filtered Out Items' that either turns off all filtered items or shows them in a lighter mode when Gantt is filtered. This feature also further aligns Simio Portal with Portal Desktop.

Simio Release 14 – Sprint 238 – April 26, 2022

Welcome to Sprint 238! In this Sprint, you notice an enhance Expression Editor as well as two other UI enhancements. There is also a new API feature that allows for object rotation and an enhancement to Step API. This Sprint wrapped up with a new Portal debugging tool.

Desktop Edition Features:
Enhanced Expression Editor – This feature was designed to improve the user experience of entering long expressions into the Expression Editor. The Expression Editor will now be a minimum of three text lines in height and can be gripped and dragged to best accommodate large expressions. Lastly, the expression editor can be pinned and unpinned to the property panel to further allow maximum flexibility. Simio will save the last used size and location of the expression editor.

Up and Down Buttons for States – This user requested feature allows you to move States up and down in the Definitions tab. Moving a State up or down will affect its location in the Watch window which can help with debugging.

Move Leftmost and Rightmost Buttons for Table Columns – This feature was implemented to improve data table structure management. You can use the Move Leftmost or Move Rightmost buttons to move a column all the way to the left or right, in its respective section, i.e., all Properties will stay together, and all States will stay together.

New Property Type and Runtime Interface Added to Step API – This new property type and runtime interface added to Step API that allows user code to read and modify an output table. This enhanced Step API capability enables output data to be passed into a custom step, modified within the step, and passed back to the model at runtime.

API Support for Object Rotation – This user requested feature allows you to alter the *Yaw* of an object in a Simio model via the API.

Export Portal Modelling Errors – This user requested feature lets you export all model error messages to a CSV file that automatically opens in order to improve the workflow of Portal modelling. Prior to this Sprint, Portal modelling errors would only appear after loading the model on to Portal, but with this feature, users can see the errors prior to uploading which allows error resolution earlier in the process.

Simio Release 14 – Sprint 237 – April 5, 2022

Welcome to Sprint 237! In this Sprint, you will notice two Constraint Logic element enhancements as well as two new JSON functions. We also added a Date Column in Trace and the ability to open unsigned files in Design Edition. This Sprint finished up with an enhancement to the Expression Editor. Portal Edition also received a new Import Status window.

Desktop Edition Features:

Unsigned Files Open in Design Edition – This user requested feature allows for Simio files that are chosen to be saved unsigned to open in Design Edition. This may be useful for when multiple authors working on a Simio file. Note: The recommended value of *Sign Files* is 'True'.

New JSON Functions – This feature adds two new JSON functions to Simio.
String.Json.Count(String, Path) returns the number of elements in an array found in a specified JSON formatted string using the given path.
String.Json.Value(String, Path) returns the string representation of a value found in a specified JSON formatted string using the given path.

Constraint Logic Element Enhancement – Allow User-Specified Owner – This feature allows a specific object to be the owner of a Constraint Logic element. The *Owner Type* may be specified as either the object associated with the constrained process step, the parent object, or as a specific object reference.

Constraint Logic Element Enhancement – Reevaluate *Number of Resources* and *Units per Resource* Expressions – This user requested feature enhances resource constraint evaluation. Prior to Sprint 237, the *Number of Resources* and *Units per Resource* properties were only evaluated at the beginning of the model run. This enhancement now has these two properties checked every time the resource constraint is evaluated.

Date Time Column in Trace – This user requested feature adds an additional column in the Trace window which will display the date and time of each action that occurs in the model run. This feature was designed to improve debugging models by adding another way to view time in Trace.

Portal Edition Features:

Scheduler Mode Import Status Window – This feature provides an additional drop-down window in Portal Scheduler Mode that shows the status of any tables that are being imported. This automatically refreshing pop-up page will show if each table is complete, cancelled, running, pending, or failed pertaining to its respective import. It will also display the last completion time of the import for each respective table.

Simio Release 14 – Sprint 236 – March 15, 2022

Welcome to Sprint 236! In this Sprint, you will be able to copy data from a Grid or Pivot Grid on both Portal and Desktop. Additionally, the Adam Optimizer has been added as another option for the Neural Network Trainer.

Desktop Edition Features:
Adam Optimizer – This Neural Network enhancement provides an additional *Optimizer Type* for the Neural Network Trainer. The Adam optimizer is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications.

Ability to Copy Data from a Grid or Pivot Grid – This user requested feature was implemented to allow grid or Pivot Grid data from Simio Desktop to be copied and then later pasted into another field so it can be filtered.

Portal Edition Features:
Ability to Copy Data from a Grid or Pivot Grid – This user requested feature was implemented to allow grid or Pivot Grid data from Portal to be copied and then later pasted into another field so it can be filtered.

Simio Release 14 – Sprint 235 – February 22, 2022

Welcome to Sprint 235! In this Sprint, you will notice two new Neural Network features as well as three new String functions. You will also now be able to reference State Columns with TableName[Index1, Index2] and set the default *Path Color* of a Link definition. We wrapped up this Sprint with enhancements to the Save Log and Starting Type. Portal Edition received a Date/Time regional update and the ability to use Dashboards that utilize the Periodic Statistics Logs. This Sprint finished with an UI enhancement to Scheduler Mode.

Desktop Edition Features:
Save Log Enhancements – This feature includes an additional column that contains whether the object's entry was added due to a save or a copy into another model as well as a change in time being recorded in UTC to local time. These changes were added to improve debugging.

Ability to Set *Path Color* of a Link Definition – This user requested feature allows you to set the *Path Color* of a link definition to something other than the default of 'SaddleBrown'.

New String. Functions – This feature provides three new String functions within Simio. String.Replace returns a copy of the specified string where all instances of a specified substring are replaced with another specified string. String.RegEx.Contains returns True if a substring matching a specified regular expression pattern is found in the specified string. String.RegEx.Replace returns a copy of the specified string where all substrings matching a specified regular expression pattern are replaced with another specified string.

Generate Training Data for Neural Network Automatically – This feature provides a tool to simplify the process of generating a specific number of training data records for a Neural Network. Prior to Sprint 235, you would manually have to change the *Replication Number* of the model and then rerun, however, this feature allows that change to be done automatically and ensures that there will be no duplicate streams run.

Neural Network Save Training Data to CSV Enhancement – This feature allows a user to export the predicted value and error value for each training data record after running a trained Neural Network Model.

Starting Type Enhancement – When a *Starting Type* of 'At the start of run, use current' is selected for a model, this feature allows the time zone in which the model should be run in to be selected. By default, this value is set to 'Device Time Zone'. This is particularly useful for models where the application is run in a different time zone than the physical location being modeled.

Ability to Reference State Column with TableName[Index1,Index2] – This user requested feature allows for you to directly reference a State Column by using TableName[Index1,Index2]. This feature was implemented to speed up models by reducing the amount of process logic needed to achieve the same functionality a different way.

Portal Edition Features:
Date/Time Formatting Enhancement – This feature adjusts the Date/Time formatting within Portal to match your regional browser settings.

Enhanced Dashboards – This feature adds the ability to have Dashboards on Portal that contain data from Periodic Statistics Logs.

Enhanced Scheduler Mode Control Properties Panel – This user requested features was designed to improve distinguishability between different sections of Properties. The Scheduler Mode Properties Panel will now have Categories for better sorting.

Simio Release 14 – Sprint 234 – January 28, 2022

Welcome to Sprint 234! In this Sprint, you will be able to use new location functions as well as set the length, width, and height of auto-created objects in data tables. We wrapped up this Sprint with a toggle to turn on/off Experiment statistical recording and a new Object Save Log. Portal Edition also received numerous bug fixes as well as a toggle to autofit Scheduler Mode table columns.

Desktop Edition Features:

Ability to Toggle Experiment Statistical Results Recording – This user requested feature was designed to speed up Experiment run time by turning off any model statistics collections. Setting *Generate Statistics* to 'False' can be useful for Experiment runs where only collecting KPIs from Responses is needed.

New Location Functions – This user requested feature was implemented to provide a new LocationAt.Cartesian(x,y,z) function that can be utilized in functions like LocationAt.Cartesian(0, 0, 0).Latitude or LocationAt.Cartesian(0, 0, 0).Longitude to get either the latitude or longitude geographic coordinates for any set of x, y, z Cartesian coordinates.

Ability to Set Length, Width, and Height of Auto-Created Objects from Data Tables – This feature allows for auto-created objects to have their *Length*, *Width*, and *Height* specified in columns of a data table. These three additional Object column properties will only appear when setting *Default Value Instantiation* to 'AutoCreateInstance'.

Object Save Log – This feature tracks the Saved Date Time, Product Version, and Edition for each object in an .spfx file. A new entry will only be added if the Product Version or Edition is different from the last entry. This feature was designed to assist in debugging and can be found by right clicking on an object in the Navigation menu.

Portal Edition Features:

Ability to Autofit Scheduler Mode Data Table Column Width – This user requested feature allows you to toggle autofitting data within data tables by right clicking on a column header in a respective table in Scheduler Mode.

Simio Release 14 – Sprint 233 – January 11, 2022

Welcome to Sprint 233! In this Sprint, you will be able to use OptQuest to optimize Table Configurations, use a table column for *Transporter List Name* as well as a number of Neural Network enhancements. We wrapped this Sprint up with Portal bug fixes and a Portal UI improvement.

Desktop Edition Features:
Ability to Export a Neural Network Model to an ONNX File Format – This feature allows you to export either an ONNX Neural Network model that you previously added to the Simio project or a selected Feedforward Neural network Model that you built and trained directly within Simio. You can then import this saved ONNX file into a 3rd party software that supports ONNX file format.

Ability to Untrain a Neural Network Model – This feature allows you to easily revert a trained feedforward model back into the initial untrained state.

***Untrained Prediction Value Expression* Property** – This new Neural Network property's expression is used as a substitute to return a predicted output value if the associated Neural Network is not trained.

Neural Network Enhancements – This feature provides some general improvements for Neural Network models. *Is Trained* and *Last Update Time (UTC)* properties provide additional information that can be useful in monitoring a Neural Network model. Additionally, in the Neural Network Trainer the default Number Hidden Nodes for an untrained feedforward neural network model has been changed from being always '10' to being the model's input layer size.

TransferNode Enhancement – This user requested feature expands the *Transporter List Name* property on a TransferNode when the *Transporter Type* is 'Select From List'. Prior to this Sprint, *Transporter List Name* had to be a List, but this enhancement now allows for it to also be table based.

Experiment Enhancement – This user requested feature allows Table Configurations to be included in an Experiment's OptQuest optimization. Setting *Include in Optimization* to 'Yes' on a Configuration will allow that multiple Configuration Table to have OptQuest use its algorithm to enhance the optimization capabilities of Simio.

Portal Edition Features:
Add Current User's Name to the List of Assigned Users in Scheduler Mode – This user requested feature was implement for when the current owner of a model assigns the model to a different user and then later wants to take the plan back. This feature also provides clarity by displaying the current user's name in the assigned user's dialog.

Simio Release 14 – Sprint 232 – December 3, 2021

Welcome to Sprint 232! In this Sprint, we improved our STEP file importing as well as improved our Error and Trace messaging. We wrapped up this Sprint with two new Location functions and with an Open Plan From Portal improvement.

Desktop Edition Features:
Improved STEP File Support – This user requested feature improves Simio's ability to import STEP files in a model.

Improved Error and Trace Messaging – This feature provides extra context for table resolution in Trace and Error messages.

Open Plan From Portal Enhancement – This user requested feature pulls in the date and time that a Portal Run has occurred when using Open Plan From Portal on Simio Desktop. This was designed as a way to verify the date and time of a Portal Run while reviewing the model in Simio Desktop.

New Location Functions – This user requested feature provides two additional location functions that can be useful in determining an object's projected location. 'Location.Latitude' and 'Location.Longitude' will provide an object's projected latitudinal or longitudinal location.

Portal Edition Features:
This Sprint contained various Portal Edition bug fixes.

Simio Release 14 – Sprint 231 – November 10, 2021

Welcome to Sprint 231! In this Sprint we added the ability to conditionally set the *Ride On Transporter* property as well as enhanced the Results tab Resource Gantt. Also, four new Add-On Process Triggers were added for Task Sequences. We wrapped up this Sprint with adding the ability to set the sequence tables will export and improving Portal data tables.

Desktop Edition Features:
***Ride On Transporter* Property Enhancement** – This customer requested feature allows you to now select 'Never', 'Always', or 'Conditional' for the *Ride On Transporter* on a Transfer Node. This feature was implemented to give additional flexibility with regards to dynamically riding on transporters.

Additional Task Sequence Add-On Process Triggers – This customer requested feature adds four additional Add-On Process Triggers in Task Sequences of Servers, Combiners, and Separators. These additional Triggers are 'Seized Resource', 'Released Resource', 'Consumed Material', and 'Produced Material'.

Results Resource Gantt Enhancement – Professional and RPS Only – This customer requested feature displays the Population Members for Workers and Vehicles on the Resource Gantt that is located in the Results tab. This was designed to improve debugging of Simio models when there is more than one population member of a transporter.

Ability to Set the Sequence Data Table Exports Occur – RPS Only – This customer requested feature allows you to define the order data tables should be exported out of Simio. The Set Export Sequence popup lives under the Export Options on a data table. This feature is useful for when data tables have dependencies on other data tables.

Portal Edition Features:
Duration Since Last Import Added for Bound Data Tables within Scheduler Mode – This customer requested feature informs Portal Schedulers the amount of time that has passed since a respective data table has been last imported.

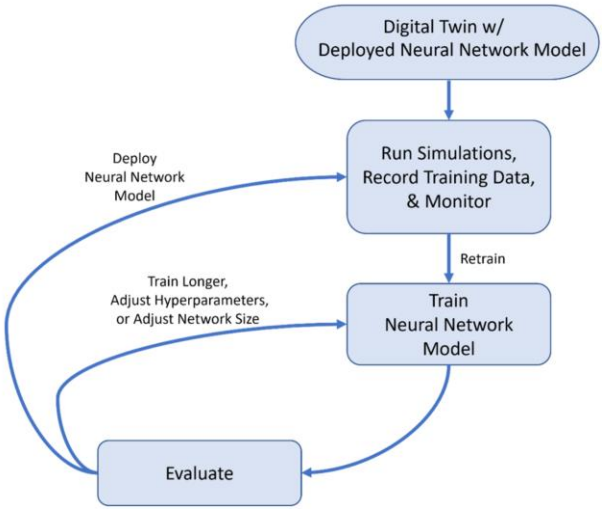
Simio Release 14 – Sprint 230 – October 18, 2021

Welcome to Sprint 230! This is the second major release of 2021. In this Sprint, we added the Neural Network Element and Neural Network Models which allows you to use Simio's machine learning features. This release also contains a SimBit demonstrating how to use the neural network features.

Desktop Edition Features:

Neural Network Models – **Academic, Professional, RPS Only** – Simio's new machine learning features allow you to not only use a neural network (NN) in Simio model logic, but you can also use the simulation to collect training data and then train your neural network from within Simio. Simio provides the ability for users to define, train, and use a NN in a Simio model with no external libraries or complex coding. Simio also supports the popular AI file format standard, ONNX. Users can import a NN into a Simio model that was defined and trained in any number of popular products which support the ONNX file format.

The Simio implementation of neural networks focuses on regression models for predicting values. These models are often feedforward, where data moves from the input neurons of the network to the output neurons and does not loop back to the input neurons. Currently, Simio supports training feedforward neural networks with one or more inputs and one output. Add a neural network to the model to make the decisions, and then we train the neural network using simulation data so that the model gets better and better at making the correct decision over time.



New SimBit: Neural Networks – This SimBit includes four models providing examples of Simio's neural network features. This SimBit starts with a base system and then shows how to set up a model to record training data, training the model, and then finishes up by demonstrating using a previously trained neural network model saved in the ONNX file format.

Portal Edition Features:

OptQuest in Portal Experimentation – Simio has added support for using the OptQuest Add-In with experimentation on the Simio Portal. This functionality will require an OptQuest license.

To learn more about experimentation on the Simio Portal product, contact your Simio representative, Simio authorized partner, or email us at info@simio.com

Simio Release 14 – Sprint 229 – September 24, 2021

Welcome to Sprint 229! In this Sprint, we have added a new SimBit, added a toggle for Breakpoints, added the ability to set the order data tables import, and provided the ability to select an active binding for a model in Portal Experimentation.

Desktop Edition Features:
New SimBit: Controlling Entity Orientation with Queue - This SimBit project includes two models providing use cases and illustrations of entity orientation in a queue.

Breakpoint Toggle – This feature allows you to make all Breakpoints visible or invisible in the Facility window by toggling the Breakpoints button on the Visibility Ribbon. Breakpoints will still exist with the toggle turned off, instead they will just not be shown.

Ability to Set the Sequence Data Table Imports Occur – This feature allows you to define which order the data tables should be imported into Simio when being auto-imported. The Set Import Sequence popup lives under the Binding Options on a data table. This feature is useful for when data tables have dependencies on other data tables.

Portal Edition Features:
Ability to Select Active Binding in Portal Experimentation – This feature allows for models that have been uploaded to Simio Portal Edition to be able to select which Binding is to be used for each Scenario when running an Experiment on the Simio Portal.

Simio Release 14 – Sprint 228 – August 31, 2021

Welcome to Sprint 228! While there are no new features added to our Desktop product this Sprint, we have added a number of features in our cloud-based Portal Edition.

Portal Edition Features:
We've added a new Portal Users Guide, which documents the user experience in our Portal Product. You will also notice an improved administrative experience on Portal, when adding new users and assigning licenses, which is now all controlled within the Users Administration page. There is a new feature which allows an administrator to enable custom text across the top of each Portal page. For example, if you have a test Portal environment and a production Portal environment, you might consider adding custom text across the top of each to indicate to your users which Portal they are currently logged in to. Finally, you will see the ability for a model owner to assign model users on their model and these users will not only be able to see the results of the plans run on that model but will also be able to copy a plan and download a plan to Desktop.

Simio Release 14 – Sprint 227 – August 10, 2021

Welcome to Sprint 227! In this Sprint, you will notice Entity Workflow Gantt enhancements as well as an Extras Library Bay enhancement. You will also notice an enhancement to the Produce step. We finished up this Sprint with support for Sketchup 2021 and a .NET Framework change.

Extras Library Enhancement – Bay *Traversal Direction*
This customer requested feature allows you to specify the direction that the bridges in the Bay move, either 'Left and Right' or 'Forward and Back'.

Installation Change – Simio Now Requires .NET Framework 4.7.2
This change requires you to have .NET Framework 4.7.2 on the machine running Simio. If you are on an up-to-date version of Windows 10 or Windows Server 2019, you will notice no difference. If you are not up-to-date, please install Simio via setup.exe which will bootstrap the .NET Framework 4.7.2 for you.

Sketchup 2021 Support
We have updated our Sketchup import to support Sketchup 2021.

Ability to Place Associated Tasks Under Owner Row on the Entity Workflow Gantt
This feature allows you to better distinguish which Task is associated with each Owner in the Entity Workflow Gantt. In Sprint 227, you can set a Log Expression Column in the Task Log that corresponds to a Log Expression Column in the Resource Usage Log and when you set the *Owner Row* of the Gantt to this column, Tasks will now appear under them.

Ability to Place Transporters Under Owner Row on the Entity Workflow Gantt
This feature allows you to better distinguish which Transporter is associated with each Owner in the Entity Workflow Gantt. In Sprint 227, you can set a Log Expression Column in the Transporter Usage Log that corresponds to a Log Expression Column in the Resource Usage Log and when you set the *Owner Row* of the Gantt to this column, Transporters will now appear in the Owner's Row.

Produce Step Enhancement
This customer requested feature for an *On Produced Process* on the Produce step allows you to specify an optional process to be executed when Material is produced.

Simio Release 14 – Sprint 226 – July 20, 2021

Welcome to Sprint 226! In this Sprint, you will now be able to import from an Excel PivotTable and see which parent object an associated object is associated to in the Internal References. We wrapped up this Sprint with added export support for additional databases Data Connectors.

Ability to Import from an Excel PivotTable
This customer requested feature allows you to import data for a data table directly from an Excel PivotTable object, not just an Excel Worksheet.

Improvement to Internal References Associated Column
This customer requested feature enhances the Associated Column in the Internal References of the model to now show what parent object the associated object is associated to and places it in the Associated To column.

Ability to Export to MySQL, Oracle, and SAP HANA with Database Data Connector
This feature request allows Simio to now support exports for MySQL, Oracle, and SAP HANA.

Simio Release 14 – Sprint 224/225 – June 29, 2021

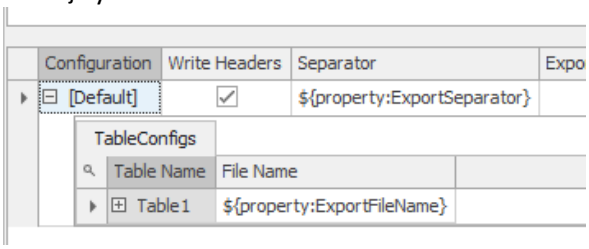
Welcome to Sprint 224/225! In this Sprint, you will notice the Standard Library and Flow Library .spfx files have moved to a new location as well as object orientation improvements. We finished up this Sprint with improvements to our importing/exporting capabilities.

File Location Change
In this Sprint we have moved the location the Simio Standard Library and Flow Library files to [Simio Install Location]\Libraries folder (likely C:\Program Files (x86)\Simio\Libraries).

Ability to have an Object's Attached Nodes, Queues, Labels Respond to Orientation Changes
This customer requested feature allows you to toggle on/off if an attached Node, Queue, or Label will rotate (change its location) with the parent object. Additionally, Labels and Queues you can toggle whether or not the orientation will change with the parent object. By default, all attached things with rotate its location and orientation with its parent. This toggle can be found by right clicking on the attached object.

Auto-Created Objects Enhancement
This customer requested feature allows you to specify an Auto-Created Object's *Yaw* property so when the object is created, it will be created with a specified orientation.

Ability to Map Property String Values to Import/Export String Parameters
This feature adds a new syntax ${property:Name} which inserts the string value of the property 'Name' in string parameters for all import/export Data Connectors. This allows you to add two new top level string properties called ExportSeparator and ExportFileName and this allows you to setup a CSV Exporter using the newe ${property:Name} syntax like so:



Once that is setup, the values given for ExportSeparator and ExportFileName will be used for the parameter values of the exporter and will live update.

Ability to Import from an Excel Table
This customer requested feature allows you to import data for a data table directly from an Excel Table object, not just an Excel Worksheet.

Simio Release 14 – Sprint 223 – May 18, 2021

Welcome to Sprint 223! In this Sprint, you will notice we have consolidated some of our file locations as well as added the ability to export the Error Log.

File Location Change

In this Sprint we have moved the location of all files that were installed to C:\Users\Public\Documents\Simio, such as SimBits, Examples, Symbols, Path Decorators, etc., to the same location we install Simio, C:\Program Files (x86)\Simio, by default.
Simio now looks in three places for things to load into Simio
1. The new standard path of Simio Install Directory\Thing, i.e., C:\Program Files (x86)\Simio\Symbols)
2. The user path of My Documents\Simio\Thing, i.e., C:\Users\MyName\Documents\Simio\Symbols
3. The legacy path of Public Documents\Simio\Thing, i.e., C:\Users\Public\Documents\Simio\Symbols
Note: If you plan on saving an older version of Simio on your computer, you will need to paste an older version of what use to be in C:\Users\Public\Documents\Simio back into that same spot *after* installing Simio 14.223.

Ability to Export the Error Log

This user requested feature was designed to allow for the Error Log to be viewed in 3rd party software when there are large volumes of data errors to analyze. When you right click on the Error Log, you can select “Export View To” and then choose either CSV or Excel.

Simio Release 14 – Sprint 222 – April 27, 2021

Welcome to Sprint 222! In this Sprint, you will be able to have more Tie Breakers for Largest/Smallest Value Selection Rules and be able to use a new 'Math.ExpectedValue' function.

New Math Function – Math.ExpectedValue(RandomExpression)
This new feature comes from an academic request. This new function deterministically evaluates the specified random expression and returns the expected value.

Expanded *Tie Breaker Rules* Functionality for Largest/Smallest Value Selection Rules
This new feature lets you use a repeat group for *Tie Breaker Rules* on 'Smallest Value First' and 'Largest Value First' so you can have multiple tiebreakers ranked in the order they are listed.

Simio Release 14 – Sprint 220/221 – April 6, 2021

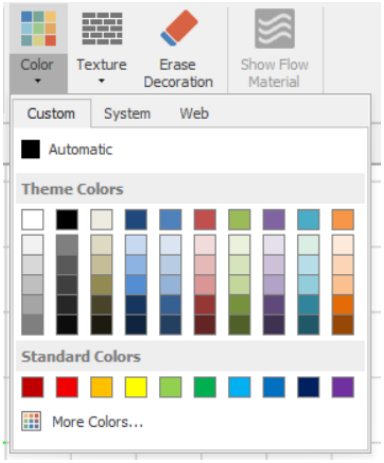
Welcome to our release of Simio 14! In this Sprint, you will be able to have entities visible in a Queue with the Queue being invisible, as well as use an enhanced color picker. We also improved our Task Sequence expression syntax. Additionally, you will be able to investigate four new SimBits!

Queue Line Externally Visible Toggle
This user requested feature allows objects to appear where an invisible Queue is located. This per-Queue toggle is accessed by right-clicking on a Queue and clicking "Queue Line Externally Visible", turning this toggle off will render the Queue invisible but objects will still appear in the Queue's location in an external space.

Improved Expression Syntax in Task Sequence *Condition or Probability*
This feature simplifies the expression needed when wanting to use a Server reference in a Task Sequence. Prior to Sprint 220, an expression had to reference the ModelEntity, for example, 'ModelEntity.Location.Parent.Server.StateVariable>1', and now you can use a simpler expression such as 'Server.StateVariable>1'.
Improved Changeover Logic
This feature was designed to improve the speed Simio runs when using a large "from-to-changeover matrix" directly in a data table that is mapped to the Changeover Logic's *Setup Transitions* repeat group.

Note: If loading a model saved in Sprint 219 or earlier, the *Always Use Property Parent Table References* advanced compatibility bit will need to be set to 'True' to use the modelling approaches illustrated above.

Enhanced Color Picker
This feature was implemented to improve the animation in Simio. This feature was intended to give you a more dynamic color selection throughout Simio as well as provide the ability for custom colors in your animation. Note: Custom colors can have transparency, however, proceed with caution as the behavior may not do exactly what you expect.



New SimBits
Constraint Logic – This SimBit contains three models demonstrating the use of Constraint Logic Elements for modelling constraint-driven situations.

Merging Conveyors Controlled By Gate – This SimBit contains three models demonstrating the use of a Resource as a gate for merging conveyors.

Transporting Varying Seized Entities – This SimBit contains two models demonstrating a modelling approach for considering an Entity's size for loading on a Transporter with limit capacity.

AutoCreate Paths With Vertices – This SimBit contains one model that demonstrates a modelling approach for auto-creating links that have vertices in them.

Simio Release 12 – Sprint 219 – February 23, 2021

Welcome to Sprint 219! In this Sprint, you will be able to place objects and graphics in Layers, turn automatic exporting of bound tables and logs on or off with a global setting, copy an Experiment from one model to another and duplicate a model in a new way.

Multi-Layer Modelling
This highly requested feature allows users to place objects, animation graphics, and symbols in different Layers. You can then render certain Layers "Invisible" allowing for objects, animation graphics, and symbols to be hidden. This feature resides on the Visibility Tab. Each object only lives in one Layer and this could be useful for highlighting certain areas of a model or modelling multi-level designs. This is a feature we intend to further improve, please give us feedback so we can have this feature better meet your needs.

Enable/Disable Automatic Bound Table and Log Exporting – RPS Only
This feature allows users to alter a global setting to turn on/off the automatic exporting of all bound tables and logs. This is best used while testing and debugging models where you wish to reduce time by not exporting tables and logs repeatedly.
Note: You can still manually export tables and logs even when this setting is false.

Ability to Copy an Experiment from One Model to Another
This customer requested feature allows users to copy an Experiment from one model to a different model. This is done by going into the Project Folder and either pressing the Copy and Paste buttons or by using Ctrl + C and Ctrl + V. After pasting, you will be able to select which model to paste the copied Experiment in to. This is useful for when you are often creating new models and want the Experiment to stay consistent through the different models.

Ability to Duplicate a Model Without Having to Go to the Project Window
This feature allows for a simpler way to copy a model's definition. This feature can be used by right clicking on the model you wish to duplicate and then selecting "Duplicate Model".

Simio Release 12 – Sprint 216/217/218 – February 2, 2021

Welcome to Sprint 217/218! In this sprint, we added the Extras Library as a third library option. Finally, we enhanced the Units Per Resource property to allow for resource-dependent values.

Extras Library
In this Sprint, you will now see a third library option below the Library objects. This is a beta version of the Extras Library. We are still fine tuning it, but at this point it is stable and ready for use. We think you will find that this library establishes an excellent foundation and framework for manufacturing modeling. Please give us your Extras Library feedback to help meet your needs.

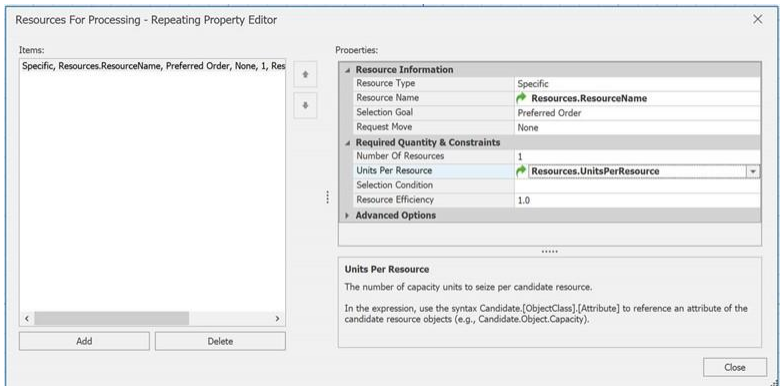
Initial Set of Extras Library Objects
Bay – A Bay defines a rectangular region over which Cranes may move.
Crane – A Crane is moved by a Cab and may pickup and drop-off entities at nodes that fall within the specified Bay.
Robot – A composite object that represents a robot. The Robot Hand is a transporter that can pickup and drop-off entities. Select the hand to edit its properties.
Rack – A Rack is used to hold entities for a specified storage time. A Rack can be used with a LiftTruck for placing items on different shelves within the Rack.
LiftTruck – A LiftTruck object may be used to transport entity objects between node locations. When used in conjunction with a Rack, the Lift on the LiftTruck will move vertically to the appropriate shelf height before executing a pickup or drop-off. Additionally, an 'On Demand' routing type LiftTruck may be used as a moveable resource that is seized and released for non-transport related tasks by model process logic.
Elevator – An Elevator is used to carry entities in a vertical direction between a collection of ElevatorNodes.
ElevatorNode – An ElevatorNode defines an entry and exit node for the entities that ride on an Elevator. Each ElevatorNode references its associated Elevator and each Elevator references a list of associated ElevatorNodes.

Enhanced Units Per Resource Logic
This customer requested feature allows the *Units Per Resource* property in Seize Step, Release Step, and Resource Availability Constraint Logic to be a resource-dependent value. This allows for easier modelling of secondary resource capacity unit requirements. For example, the requirements might be 4 capacity units if using Resource1, 8 capacity units if using Resource, or 12 capacity units if using Resource3.

Resources

	Resource Name	Units Per Resource
1	Resource1	4
2	Resource2	8
3	Resource3	12
*		

Now, if using a data table to define the resource list members, the *Units Per Resource* property may be specified as another column in the table.

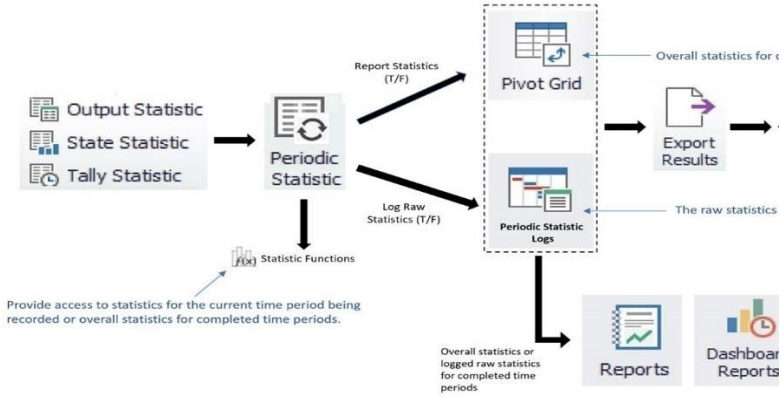

Resources For Processing - Repeating Property Editor
Items:
Specific, Resources.ResourceName, Preferred Order, None, 1, Res
Add
Delete
Properties:
Resource Information
Resource Type
Specific
Resource Name
Resources.ResourceName
Selection Goal
Preferred Order
Request Move
None
Required Quantity & Constraints
Number Of Resources
1
Units Per Resource
Resources.UnitsPerResource
Selection Condition
Resource Efficiency
1.0
Advanced Options
Units Per Resource
The number of capacity units to seize per candidate resource.
In the expression, use the syntax Candidate.[ObjectClass].[Attribute] to reference an attribute of the candidate resource objects (e.g., Candidate.Object.Capacity).
Close

Simio Release 12 – Sprint 214/215 – December 11, 2020

Welcome to Sprint 214/215! In this sprint, we added a Periodic Statistic Element, additional Skyboxes, an additional export option for Data Tables, and Table Schemas for Elements, Objects, and Repeat Groups. We also removed DirectX9 support from the software and added support for Sketchup 2020. We improved our dynamic selection rule at a Routing Group element to dynamically select entity-destination assignments. Lastly, we added the Extras Library along with SimBits and an Example to the software.

Periodic Statistic Element

This highly requested feature allows users to collect periodic statistics on an Output Statistic, a State Statistic, or a Tally Statistic. This Periodic Statistic allows users to specify the *Period Length* as well as the *Number Of Periods* as well as if the period repeats. Users with Professional or RPS Edition software will also have access to three new logs associated with the new Periodic Statistic element: Periodic Output Statistic Log, Periodic State Statistic Log, and Periodic Tally Statistic Log.



Extras Library: Library and SimBits Installed with Simio

The Extras Library, previously available on the Simio User Forum, is now installed automatically and can be found in the Examples folder, typically in the following directory: C:\Users\Public\Documents\Simio\Examples\ExtrasLibrary. The Extras Library currently includes 8 pre-built object definitions that can be used to model material handling systems. Two objects, the Bay and the Crane, were designed for modelling multiple cranes operating simultaneously in a bay. Three objects, the Elevator, the ElevatorNode, and the ElevatorSelectorNode, were designed for modelling elevators. Two objects, the LiftTruck and the Rack, were designed for modelling entity storage and retrieval. Finally, the Robot object was designed for modelling entity movement between locations via a revolute robot. A new example and several new SimBits were added to provide examples and use cases for these objects.

New SimBit – Periodic Statistics

This SimBit includes three models which show how to collect hourly statistics for entity time in system, resource state over three-shift days, and daily average Scheduled Utilization over a week. These SimBits utilize the Periodic Statistic Element which allows users to collect periodic statistics on an Output Statistic, a State Statistic, or a Tally Statistic.

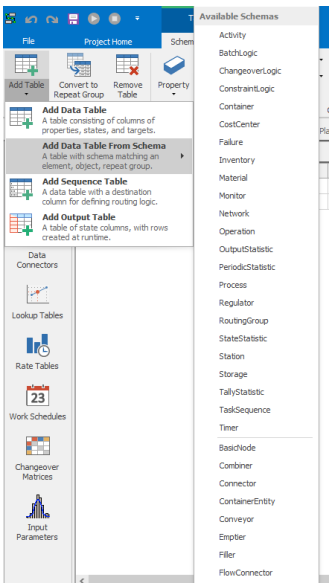
New Example – Engine Repair Using Extras Library

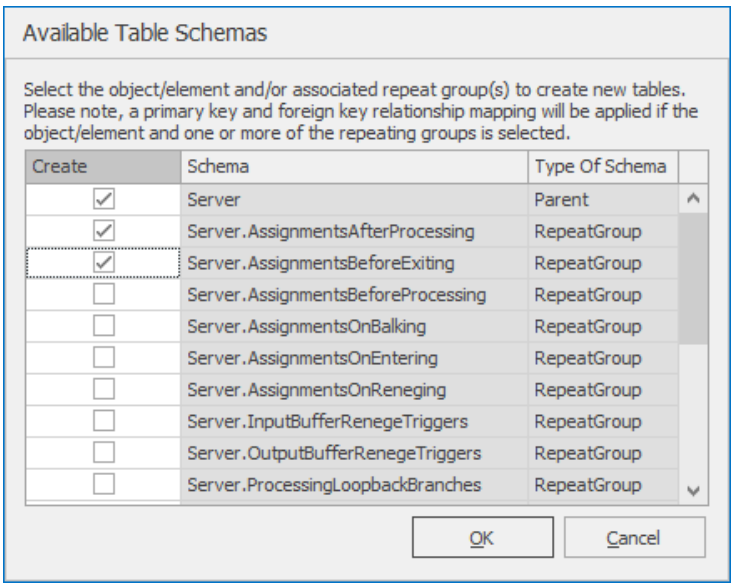
This model represents an engine repair line. Engines requiring repairs arrive from a truck dock and are placed in a holding area. Cranes will move these engines from station to station. The first stop is an uncasing station that removes the engine's outer case. The engine then moves to an inspection station where diagnostics are run, and the parts needed for repair are recorded. After, the engine will move into a middle holding area and then to a repair station when one is available. At the repair station, kits with the required parts are delivered and the engine is repaired. Once repaired, if there is room in the middle zone, the engine will move there, then to travel straight to testing, else it will be moved into one of the empty spaces at the end of the line to wait. Repaired engines will be tested at the test station, re-cased at the re-casing station, and then moved out for shipping at the truck dock.

The parts needed for engine repair are pulled from containers which sit on pallet racks in the storage area. A forklift will pick up the container off a shelf and move it to a drop off zone. In this building, the storage facility is located on a different level. Once dropped off by a forklift, the storage container must ride down an elevator to the lower level. From the elevator, a worker moves the container to a conveyor system. This conveyor system moves the containers past robots which will pick up the needed parts out of the container and then place the part in a small kit bin. At the end of the container conveyor, new parts are added back into the storage container and sent up to the upper level on another elevator where they are then returned to the shelf by a forklift. At the end of the kit conveyor, a worker will deliver the kits to the engine repair stations.

Added Table Schemas for Elements, Objects, and Repeat Groups

This internally requested feature was added to the software to make the auto-creation of Elements, Objects, and Repeat Groups easier. These schemas can be added into a model by selecting "Add Data Table From Schema" and then by checking the schemas you want to create.





DirectX 9 Support Removed from Simio
This user requested feature was implemented in order to stay up to date with security measures and any users that have previously chosen DirectX 9 in the past will now use DirectX 11 starting in Sprint 214.

Sketchup 2020 Support
We have updated our Sketchup import to support Sketchup 2020.

Added Additional Skyboxes
We have added five new Skybox graphics: Airport, CleanMFGFacility, Hospital, MFGPlant, and Warehouse. Skyboxes can be used to make a better-looking model horizon – these are useful not only for outdoor applications, but also indoor applications (e.g., for showing walls of a building).

Added Additional Export Option for Data Tables: Export at Experiment Replication End – RPS Only
This customer requested feature allows data to be exported at the end of each Experiment replication run.

Simio Partner Edition
Authorized Simio partners will see new features available in this release.

Improved Dynamic Selection Rule at a Routing Group
This feature allows a new dynamic selection rule algorithm implemented by a routing group will execute as follows:

- Choose the best entity-node assignment from all possible entity-node pairs using the dynamic selection rule priority.
- The tiebreaker between entity-node assignments with equal dynamic selection rule priority is the route request queue rank.
- The secondary tiebreaker between entity-node assignments with equal dynamic selection rule priority that involve the same entity is the node selection priority as determined by the *Selection Goal* property on the Route step or TransferNode object.

Note: If loading a model saved in Sprint 214 or earlier, the *Allow Cross-Node Comparisons When Using Routing Group Dynamic Selection Rules* advanced compatibility bit will need to be set to 'True' to use the modelling approaches illustrated above.

Simio Release 12 – Sprint 213 – October 13, 2020

Welcome to Sprint 213! In this sprint, we enhanced file merging, Controls, and Experiments.

Improved File Merging – Opt-out of Signing Files – Professional and RPS Only
This user-requested feature allows users to opt-out of signing files when they save them via the *Sign Files* property on the project. This feature was designed to improve multi-author file merging under source control/simproj scenarios. Signing files is recommended, as it allows certain lower editions of Simio to open files. However, for multi-author scenarios, where multiple users are working on the same Simio project (via source control around a simproj format), setting *Sign Files* to 'False' will help avoid conflicts around file merging.

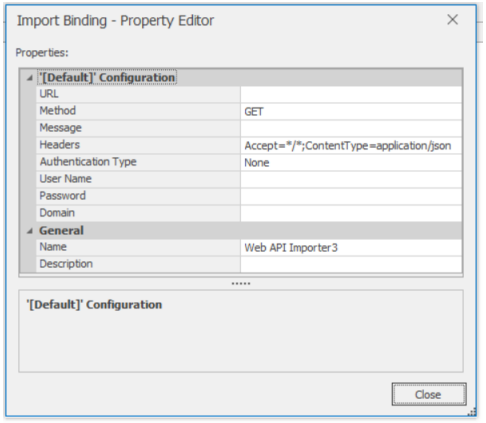
Control Value Enhancements – Ability to Update all Properties' *Default Value* to Match the Current Control Value
This user-requested feature allows users to save all current Control values to become the *Default Value* of the properties they correspond with. This can be done by right clicking in the Controls area and selecting "Save All Current Values As Default".

Experiment Enhancements – Ability to Copy the Model's Control Values to this Scenario
This user-requested feature allows for Scenarios within an Experiment to have its Controls set by what the current values of the Controls in the model are instead of the *Default Values* of these Controls. This right click on the desired Scenario reduces the need for manually typing in Control values that match the values that were last used in Interactive Mode.

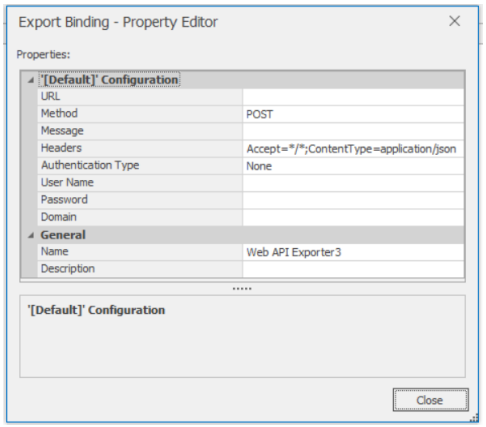
Simio Release 12 – Sprint 212 – September 22, 2020

Welcome to Sprint 212! In this sprint, we added an additional Data Connector Importer and Exporter.

Web API Data Connector Importer
This feature was requested by customers and consultants to import data from various HTTP endpoints, mostly web APIs. This data is generally returned in either JSON or XML format and this feature allows that data to be put into our Tables.



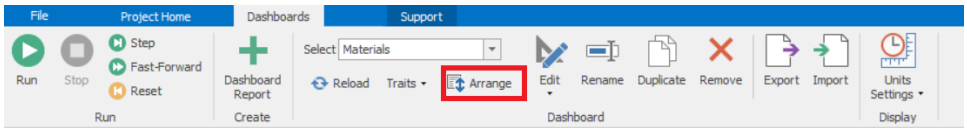
Web API Data Connector Exporter – RPS Only
This feature was requested by customers and consultants to export data to various HTTP endpoints, mostly web APIs. This data is generally assumed to be exported in a row-cetric format, that is, one HTTP call per row of data to export.

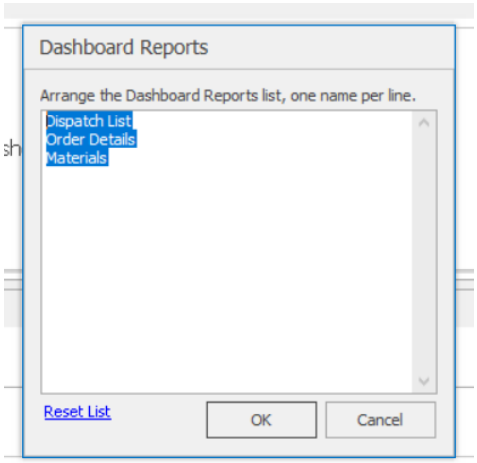


Simio Release 12 – Sprint 211 – September 1, 2020

Welcome to Sprint 211! In this sprint, we enhanced our Constraint Logic element, our Documentation Report, and the Experiment logic. Additionally, Simio now supports additional types of file imports. We finished up this Sprint with improved functionality on a bound table and the ability to reorder Dashboards and Table Reports.

Ability to Reorder the Order that Dashboards and Table Reports Appear
This feature was implemented to provide a way to define the order that Dashboard and Table Reports appear in the dropdown.





Constraint Logic Enhancement – *Constraint Scope*

This feature was designed to improve the speed of the Constraint Logic element. The *Constraint Scope* property determines the scope of the constraint when it is applied to a request. This feature allows users to specify if the constraint will be checked at most once, 'Request', or if the constraint will be checked for each destination, 'RequestedItem'.

Experiment Enhancements
This feature was designed to improve the speed of Experiment runs. The *Concurrent Replication Limit* default of '0' will be set to physical cores when running locally.

Documentation Report Enhancements
This customer requested feature was designed to provide more information in the Documentation Report. This enhancement now allows the *Description* of a Table to appear in the Documentation Report.

Ability to Add and Remove Columns and Set Primary Key on a Bound Table
This feature was implemented to allow users to add and remove property columns and set a property column as a Column Key while the table is bound. Prior to Sprint 211, users would have had to unbind the table, make changes, and then rebind.

Constraint Logic Enhancement – Provide Option that will Allow Switch Resource if Possible to Happen Before Constraint Logic Elements in the Simio Event Calendar
There is a new compatibility bit "Schedule Constraint Logic Related Queue Checks As Latest Current Events". Default value of this setting is 'False' for models built prior Sprint 12.211, 'True' for new models built in Sprint 12.211+. When set to 'True', Simio will schedule constraint logic related queue checks as latest priority current events, thereby allowing queue checks scheduled for any other reason to happen first.

Ability to Import Additional File Types – STEP Files and glTF Files
This user requested feature was implemented to allow for Step and glTF files to be imported in Simio. The new supported file extensions are as follows:

- STEP - .ifc, .ifczip, .stp
- glTF - .gltf, .glb

Simio Release 12 – Sprint 210 – August 11, 2020

Welcome to Sprint 210! In this sprint, we enhanced our CSV and Excel Data Connectors and we enhanced our Constraint checking.

Enhanced CSV and Excel Data Connectors
This customer requested feature allows for the *Import Folder* property of the CSV Importer and the *File Name* property to be set via the File Explorer. This feature eliminates the need to manually type in the path needed to reach a file and replaces it with the ability to click through a file explorer to do so.

Enhanced Constraint Checking
This feature was implemented to provide a more complete ConstraintLogic element modelling approach. *Exhaustive Constraint Checking*, a property located in a model's Advanced Options Property

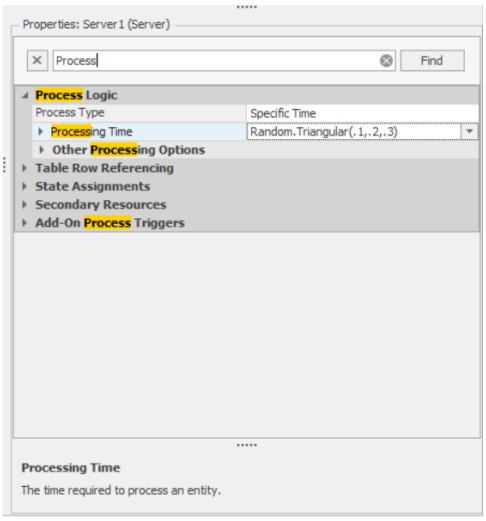
Pane, provides the option to check Constraint Logic elements after any relevant state change in the system. Setting this property to 'True' will provide the most accurate detail in Trace and the Constraint Log but may result in longer run times. Setting this property to 'False' will stop the checking when a Constraint is met.

Simio Release 12 – Sprint 209 – July 21, 2020

Welcome to Sprint 209! In this sprint, we have enhanced our Documentation Report, enhanced our Active Binding Importation/Exportation, updated our Property Grid, and added another *Constraint Type* to the Constraint Logic Element. We finished up this Sprint by updating some of our scheduling examples and adding a new String.FromList function.

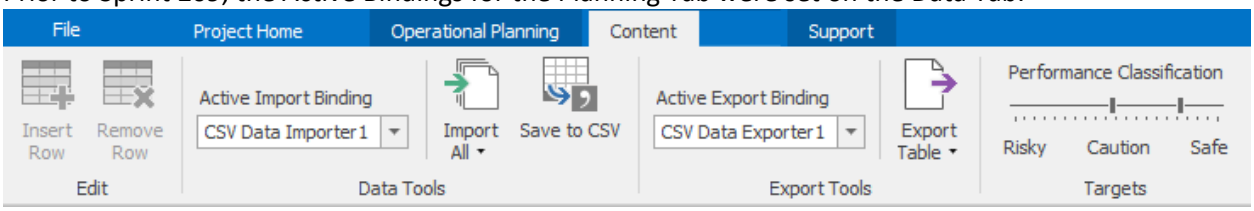
Documentation Report Enhancements
This customer requested feature was designed to provide more information in the Documentation Report. This enhancement now allows the *Description* of an Object or Element to appear in the Documentation Report.

Property Grid Enhancements
This update was implemented to keep our software as current as possible. This feature allows for the Property Grid to be searched by property (i.e. *Processing Time*) by right clicking in the Property Grid and then selecting “Show Find”.



Active Import and Export Binding Dropdowns on Planning Tab – RPS Only

This feature allows users to set the Active Import Binding and Active Export Binding on the Planning Tab. Prior to Sprint 209, the Active Bindings for the Planning Tab were set on the Data Tab.



Constraint Logic Element: *Constraint Type* 'Condition Expression'

This feature was designed so users can enforce additional constraint conditions on an applicable process step. For example, a resource cannot be seized at a Seize step or a destination node assigned at a Route step unless the requested item is available plus the additional constraint logic satisfied. This feature is an expansion on the Constraint Logic Element: *Constraint Type* 'Material Availability' from Sprint 203 and the Constraint Logic Element: *Constraint Type* 'Resource Availability' from Sprint 206.

By default, a 'Condition Expression' constraint will be automatically checked whenever the Constraint Logic is checked for the related queue item waiting in the resource's Allocation Queue or Routing Group's Route Request Queue. The Condition Expression is evaluated in the context of the constrained entity execute the Seize or Route.

To perform additional monitoring of the constraint, you can use the Monitoring Event Name property to trigger checks of the Condition Expression whenever the specified event occurs. If it is determined by one of those extra checks that the condition result has changed from false to true, then that will in turn automatically trigger a new check of the queue containing the related queue item.

For any constraint defined using a Constraint Logic element, customized values can be written to the Constraint Type, Constraint Item, and Constraint Description fields of the constraint's record in the Constraint Log (and thus displayed in the constraint's tool-tip text in the Gantt).

Updated Scheduling Examples and Templates

The ISA95 templates and SchedulingDiscretePartProduction example have been updated to include:

- *Expected Operation Time* of SchedServer and *Sequence Expected Operation Time* of SchedTransferNode fixed for use with various sequence-based scheduling rules
- Consume/Produce processing steps within SchedServer times reduced to .1 hour for demonstration of feature, *Quantity* simplified using *Skip Requirement If* for those processing tasks (Note: *Quantity* change also made to MaterialAvailability ConstraintLogic element)
- Default Entity *Initial Desired Speed* set to 'Infinity' for movement between objects
- *Dynamic Label Text* added for graphics for DefaultEntity and SchedWorkers
- Moved 'ObjectType' column in Resources table closer to ResourceName/Description

The SchedulingLaborEfficiencies example has most of the above changes, as well as:

- New ConditionExpression type constraints for both WIPDestinations and ValidDestinations have been replaced SchedTransferNode *Selection Condition.* These also include enhanced logging.
- LaborConstraint type ConstraintLogic elements contain enhanced constraint logging.

We will be making similar updates to the other Scheduling examples and templates over the next several sprints.

String.FromList Function

This customer requested feature returns the string representation of a specified integer constant in the specified string list. For example, if you have defined the following string list which contains entries "Green" = 0, "Red" = 1, and "Blue" = 2, then 'String.FromList(StringListName, 2)' will return 'Blue'.

Simio Release 12 – Sprint 208 – June 30, 2020

Welcome to Sprint 208! In this sprint, we have enhanced our Travel step, added the ability to set Link properties as a Reference Property, and implemented a feature that allows table row references that are set on a Token to be referenced in Queue Ranking Expressions. We also added additional exportation options for Tables and Logs.

Travel Step Enhancement – Zero-Time Free Space Travel

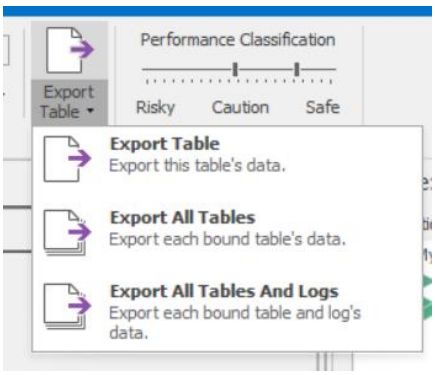
This academic feature request was designed to provide an option for zero-time transfer without a Connector. Prior to Sprint 208, the *Initial Desired Speed* of an Entity could not be set to 'Infinity' but now with this feature implementation it is possible to have instantaneous travel with the Travel step.

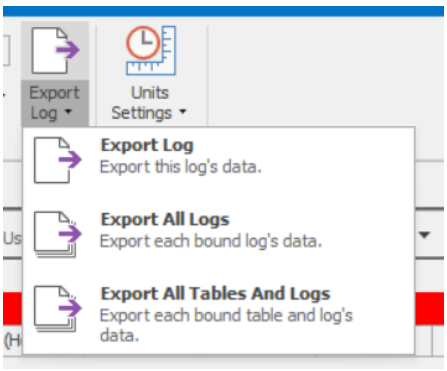
Ability to Set a Link's *Type*, *Drawn To Scale*, and *Logical Length* as a Reference Property

This feature was requested by multiple Simio users and allows for additional Link properties (*Type, Drawn To Scale,* and *Logical Length*) to be set as a Reference Property. This also allows for these properties to be set via a Data Table and altered with in an Experiment.

Ability to Export All Tables or All Logs – RPS Only

This feature provides an additional option for exporting Tables and Logs within Simio. Prior to Sprint 208, if a user wanted to export all the Tables or Logs at once, they had to do both simultaneously. This feature provides additional options to export all Tables or all Logs individually.





Allow Token-Based Table Row References in Queue Ranking Expressions

This feature allows for table row references that are set on a Token to be referenced in the *Ranking Expression* property of a resource or the *Allocation Ranking Expression* of Material or Inventory elements. This enhancement was implemented to allow behaviors such as ranking by demand priority in an Allocation Queue or ranking requests to seize a resource by processing task priority in Task Sequences.

Note: If loading a model saved in Sprint 207 or earlier, the *Allow Token-Based Table Row References In Queue Ranking Expression* advanced compatibility bit will need to be set to 'True' to use the modeling approaches illustrated above.

Simio Release 12 – Sprint 207 – June 9, 2020

Welcome to Sprint 207! In this sprint, we have added two new functions as well as enhanced two of our functions that deal with randomness.

RandomRow/RandomValue Enhancements
This academic request was implemented to provide the ability to better control randomness at different stages of a model. Both RandomRow and RandomValue functions now support an optional stream argument.

Syntax like 'TableName.ColumnName.RandomRow(2)' or 'TableName.ColumnName.RandomValue(2)' will specify random number stream 2.
Note: Using 'TableName.ColumnName.RandomRow' or 'TableName.ColumnName.RandomValue' will continue to use the default random number stream 0.

ColumnForName Function
This feature was designed to provide a simpler way to reference characteristics of a data table. ColumnForName returns the 1-based index of the table column that is the specified property name. For example, syntax like
'PressValidResources[PressValidResources.MaterialName.RowForKey(Materials.MaterialName), PressValidResources.ColumnForName(Candidate.Node.AssociatedObject.Name)]' to lookup in 'PressValidResources' table the value in the row containing the key value 'Materials.MaterialName' and the column named 'Candidate.Node.AssociatedObject.Name'.

String.IndexOf Function
This customer requested feature allows Simio users to find the character position where a string occurs within a string. String.IndexOf(string, substring) returns the one-based index of the first occurrence of a specified substring within the string.

Simio Release 12 – Sprint 206 – May 19, 2020

Welcome to Sprint 206! In this sprint, we added another *Constraint Type* to the Constraint Logic Element. There were also enhancements made to our Entity Workflow Gantt.

Entity Workflow Gantt Enhancement – Place Associated Constraints Under Owner Row – RPS Only
This internally requested feature was added to have Constraints associated with a specific Entity placed in a row below the owner Entity. Prior to Sprint 206, all Constraints would be placed in one row at the bottom of the high-level Entity. This feature allows users to see both the Owner Row and Constraint simultaneously without having to scroll through the Entity Workflow Gantt.

Note: The following three conditions must be true for this to happen:
- The Constraint Log and Resource Usage Log both have expressions with the same name
- The Resource Usage Log expression is marked as "Show in Drop Down"
- The Resource Usage Log expression is currently selected as the "Owner Row"

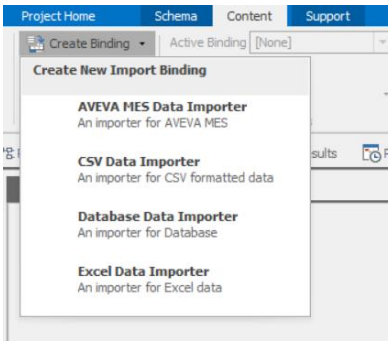
Constraint Logic Element: *Constraint Type* 'Resource Availability'
This feature was designed so users can enforce additional resource availability constraints on a Seize and/or Route Request. This feature also allows users to track unsatisfied resource availability constraints in the Constraint Log. This feature is an expansion on the Constraint Logic Element: *Constraint Type* 'Material Availability' from Sprint 203.

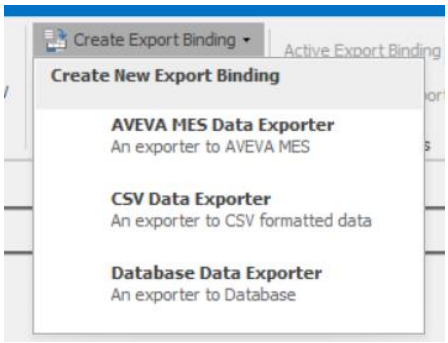
If the *Constraint Type* is 'Resource Availability', then that resource requirement will be automatically reserved using the specified selection goal when a resource is successfully seized (Seize step) or when the Entity is successfully assigned a Destination Node (Route step). This is done to ensure that other seize or route requests waiting for the same resources do not violate any resource availability constraints.

Welcome to our release of Simio 12! In this sprint, we added a new Flow SimBit, added the ability to edit an existing federated data source in Dashboards, added an enhancement to the Changeover Logic element, and added the ability to add a Data Connector from the Create Binding dropdown.

Ability to Add a Data Connector from Create Binding

This feature was added to give users more flexibility on the way they create a Data Connector, both Importers and Exporters.

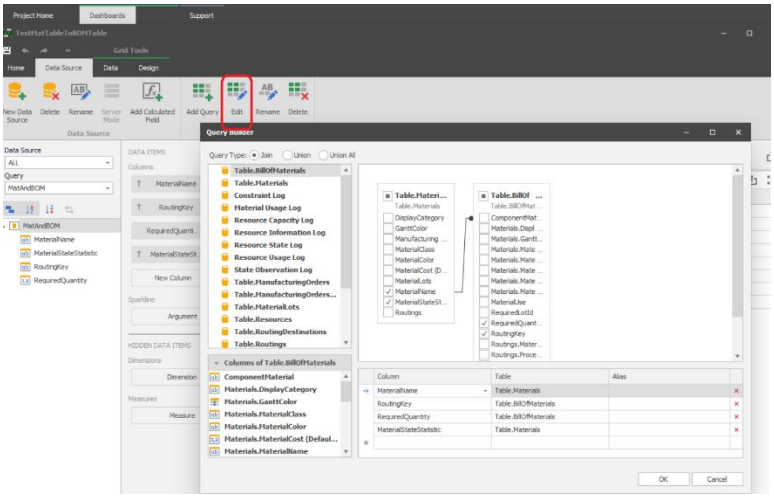




After selecting the Data Connector, the user will be able to set up the Default Configuration in a pop-up property window and then the Data Table properties can be set up either via the Data Table itself or by the Data Connector View.

Ability to Edit an Existing Federated Data Source in Dashboards

This feature request came from a customer that wanted to edit a federated data source once it already existed in a Dashboard. Previously, it was not possible to edit these data sources once they existed. A new Edit button is located on the Data Source ribbon of the Dashboard window, clicking this button allows users to edit their query.



New SimBit - TrackingTankContentsByProductTypeUsingTableStates

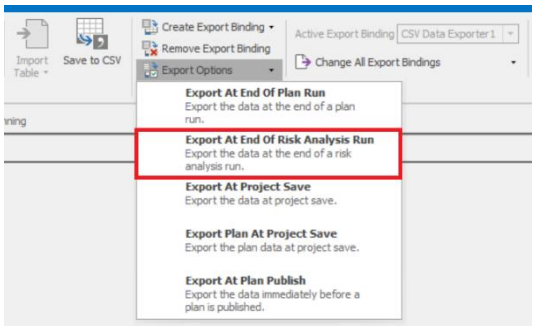
This model shows one way to keep track of individual quantities of flow volume within a Tank during a simulation run. A data table comprised of a Key Column of each Entity and Level State Columns for each Tank in the system can keep record of flow quantities by Entity Type and Tank with the use of Add-On Process logic and Monitors.

Changeover Logic Enhancement
This customer requested feature allows for users to reference either a Resource or an Entity when evaluating the *Operation Attribute* of a Setup Transition. Prior to Sprint 205, it was only possible to reference an Entity.

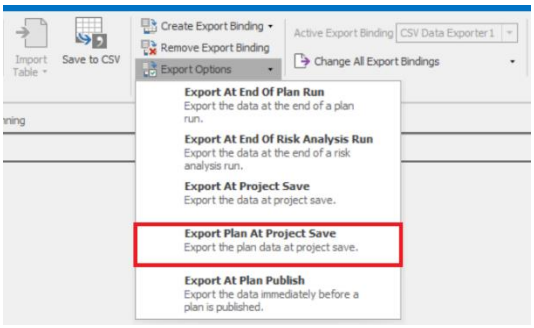
Simio Release 11 – Sprint 204 – April 7, 2020

Welcome to Sprint 204! In this sprint, we have enhanced our Data Connectors, API, and model protection capabilities. Two new export trigger options on tables were added to give users more flexibility with data exportation. We also added the ability for Targets to be exported using an Exporter Data Connector. The ability to set an Active Configuration in an Experiment was implemented in this sprint. We wrapped up this sprint with the deprecation of the XML Transformation classic binding and the conversion of the Excel classic binding to an Excel Importer Data Connector. This sprint concluded with the addition of a new example, InfectiousDiseaseSampleModel.

New Export Trigger Options on Tables – RPS Only
Export at End of Risk Analysis Run – This new export option allows for the export of data tables at the end of a risk analysis run. This feature was added to aid in the ability to export Targets.

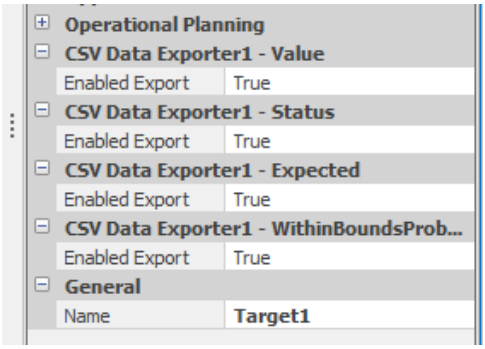


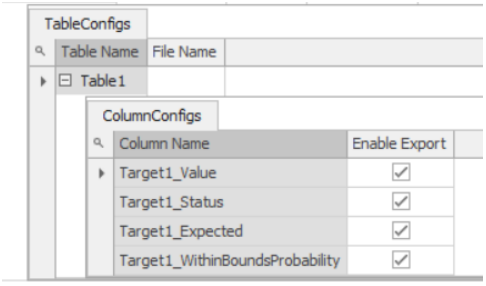
Export Plan at Project Save – This new export option allows for the export of planning tables upon saving the project. This feature was added to allow for the exportation of plan state values. These are the values shown on the Tables view of the Planning tab. The “Export At Project Save” export option exports the interactive state values. These are the values shown on the Tables view of the Data tab.



Export of Targets via Exporters – RPS Only

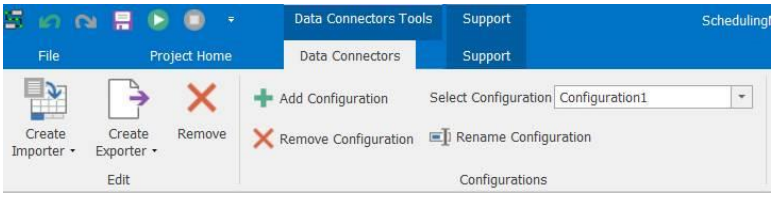
Target columns on tables now will export, just like property and state columns. This feature was added to complete the exportation of data tables with Exporters. Note: Interactive gives two columns: Value and Status. Planning gives four columns: Value, Status, Expected, and Within Bounds Probability. Regardless of exporting location (interactive or planning), all four columns will be seen in the property grid – all four of these columns have the Boolean property *Enabled Export.* This property can also be set on the Data Connectors view.





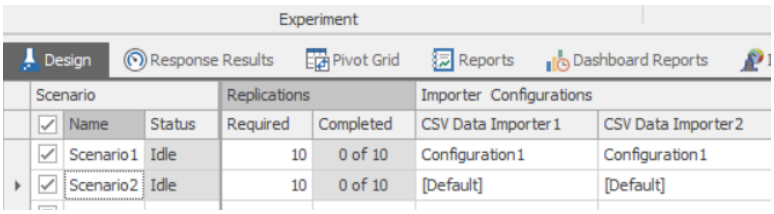
Ability to Rename a Data Connector Configuration

This feature was added to allow for Configurations to be renamed after they have been created. Note: It is not possible to rename the [Default] Configuration.



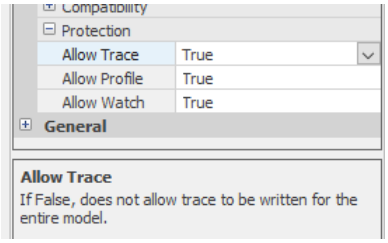
Ability to Set Active Configurations in Experiments - Importers

If an Importer Data Connector has any custom Configurations defined, that Data Connector will appear in the Experiment design view. This allows for the Configuration to be changed with ease and for different Scenarios to have different Configurations defined.



Enhanced Protection Capabilities for Models – Ability to Protect Profile, Trace, and Watch – RPS and Professional Only

In Sprint 202, we provided the capability to protect specified views within a Simio model. This enhancement was designed to provide more protection flexibility by allowing users to protect specific windows: Profile, Trace, and Watch. This model property is set via the Project Folder in the Navigation Pane, then the Advanced Options property group Protection.



API Enhancement - Ability to Publish Plan to Portal Without Opening Desktop Software – RPS Only

This user-requested feature was added to allow for the running and publishing of a plan to Simio Portal without any human participation. Prior to this feature, to publish plan results to Simio Portal a user had to open Simio and select “Publish Plan Results to Website”. This feature allows users to publish to Simio Portal in the API.

Note: There is a new SimioServerConnector.dll available as part of the main Simio install. It and its associated dependencies are found in the ServerConnector subdirectory of the main Simio installation directory. The Simio Server Connector Reference Guide can be found in the main Simio installation directory.

Deprecation of XML Transformation Classic Binding

This classic binder was deprecated due to the recent implementation of Data Connectors in Sprint 202. These Data Connectors provide more flexibility to users while also eliminating the need for classic bindings. Models created with classic XML bindings will still use the classic binding for binding to external data. Models created in 204 or later can show the deprecated table bindings if the setting Show Obsolete Table Bindings is set to ‘True’. This property is located by going to File -> Settings -> Add-ins.

Conversion of Excel Classic Binding to Excel Data Connector – Importer Only

This feature was added to complete the transition of using classic bindings to the new Data Connectors, which was implemented in Sprint 202. This new Importer allows for the importation of Excel data to a table via a Data Connector. Models created with classic Excel bindings will still use the classic binding for biding to external data. Models created in 204 or later can show the deprecated table bindings if the setting Show Obsolete Table Bindings is set to ‘True’. This property is located by going to File -> Settings -> Add-ins.

New Example – Infectious Disease Sample Model

During the time of an infectious disease outbreak, the strain on hospital and medical care networks is unprecedented. There is uncertainty in what demand a healthcare facility might see. In response, the Simio team has created a model to assess the capacity of critical resources for a hospital. We hope this model facilitates an assessment of a hospital network's resources and provokes discussion if action needs to be taken.

A healthcare provider's ability to combat an infectious disease is directly impacted by their quantity of resources available. The accessibility of beds and machines such as ventilators are crucial factors for patients that require intensive care. Being able to anticipate the demand for these resources in a facility is vital with repercussions linked to the mortality rate. With our model you will be able to experiment with the number of beds, ventilators, and personal protective equipment (PPE) such as masks, gloves, and gowns, to find the level necessary to provide care to the infected population. This model may be useful as is or used as a starting point to customize a model to meet your specific needs.

Simio Release 11 – Sprint 203 – March 17, 2020

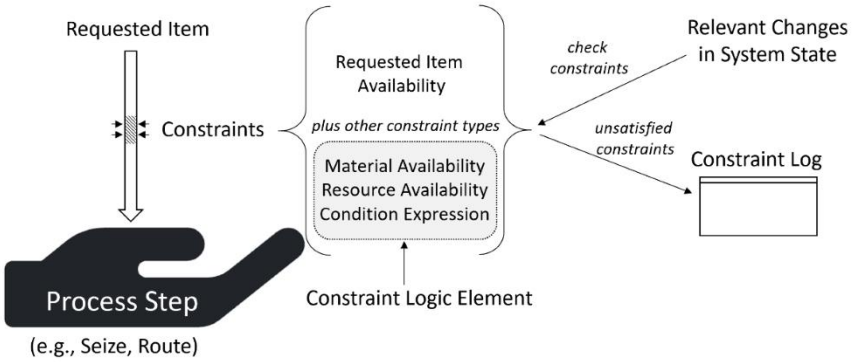
Welcome to Sprint 203! In this sprint, we added a new Constraint Logic element to allow users to put material availability constraints on Seize and Route Requests. We also enhanced the Watch Window to watch "sub-objects" of objects. The CSV and Database bindings were deprecated as a result of the Data Connectors implementation in Sprint 202. The Wonderware Add-In was also deprecated. A new scheduling example was added and the preexisting scheduling examples and templates were updated. To finish up this release, we allowed Logs to be bound to an Exporter and added some enhancements to our importation/exportation of data.

Watch Window Enhancements
This feature was created to give more insight in to an object that is being watched during a simulation run. With this feature, it is now possible to watch "sub-objects" of objects in the Watch Window. For example, if a custom object includes a Resource object, when the custom object is placed into the Facility View and then watched, upon expansion of the custom object in the Watch Window, the Resource will now appear under the custom object.

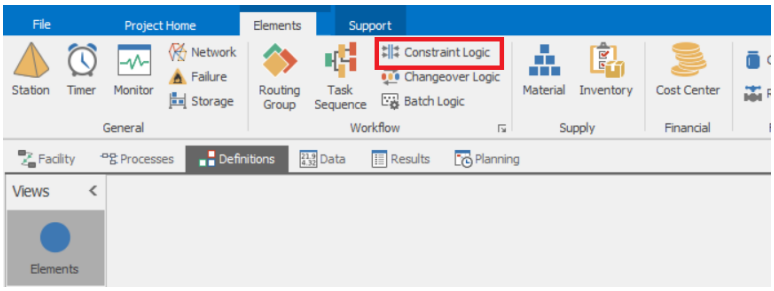
Constraint Logic Element: *Constraint Type* 'Material Availability'
This feature was designed so users can enforce additional material availability constraints on a Seize and or Route Request:

- On a TransferNode – Allows users to enforce additional material availability constraints on an entity Route Request. The TransferNode object has a new property, *Route Constraint Logic*, under the Other Routing Out Options category. This repeat group property allows for multiple Constraint Logic elements to constrain an entity's route.
- On a Server, Combiner, Separator, Filler, and Emptier – Allows users to enforce additional material availability constraints on an entity that is waiting in an object's Input Buffer and is attempting to seize that object. This repeat group property *Seize Constraint Logic* is located under each object's Other Processing Options category and allows for multiple Constraint Logic elements to constrain an entity's request to seize an object.

The figure below illustrates that a resource cannot be seized, or a destination node cannot be assigned unless that requested item is available plus the additional constraint logic is satisfied.



The Constraint Logic Element is located on the Elements view of the Definitions Tab.



Due to the creation of the Constraint Logic element, enhancements were made to both the Seize step and the Route step. These steps now have an Advanced Property *Seize Constraint Logic* and *Route Constraint Logic*, respectively, where a user can specify additional constraints on the request by referencing Constraint Logic elements.

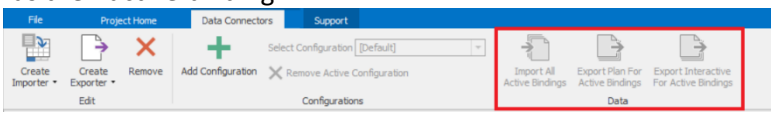
If the constraint type is MaterialAvailability, then the quantity of the material that is required will be automatically reserved when a resource is successfully seized (Seize step) or when the entity is successfully assigned a destination node (Route step). This is done to ensure that other seize or route requests waiting for the same material do not violate any material availability constraints.

Note: Any existing *Required Materials* repeat group on the Route step and the *Other Routing Out Options -> Required Materials* repeat group on the TransferNode will be deprecated and can be replaced with the Constraint Logic element. If using the deprecated method of constraining routing is desired, set File -> Settings -> *Display Deprecated Properties In Properties Window* to 'True'.

Export of Logs with an Exporter – RPS Only

This feature was designed to allow users to bind a Log to an Exporter, similar to binding a data table to an Exporter. To use this feature an Exporter must first be created, then to bind the Exporter, go to Results -> Logs -> Create Export Binding. Note: Currently to edit the binding, it must be done via the Data Connectors view.

Import/Export by Binding Type – Import Professional and RPS Only, Export RPS Only
This feature was implemented in order to have more selective imports of data. In the Data Connectors view, it is now possible to select a Data Connector and import or export all data tables that are using that Data Connector as their active binding.



New Scheduling Example
SchedulingBicycleAssembly_UpdateCreateMOandPOs – We have added a new scheduling example to our Support ribbon > Examples area that is similar to the SchedulingBicycleAssembly example. This example, however, uses Sales Orders from a data table and generates associated Manufacturing Orders (MO) and Purchase Orders (PO) based on Bill of Materials structure. Both MO and PO are tracked via output tables.

Updated Scheduling Examples and Templates
The ISA95 and WWMES templates, as well as all scheduling examples, have been updated to use the new Constraint Logic element for material availability.

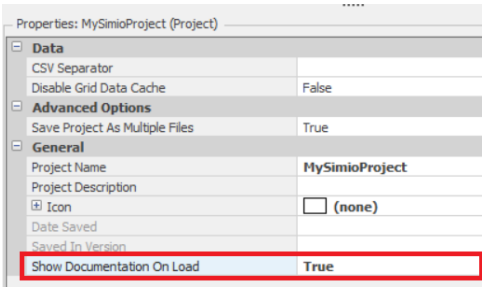
Deprecation of CSV and Database Classic Bindings
In Sprint 202, Data Connectors were introduced as a way to import and export data from a multitude of sources with the intent of giving users more flexibility. Using Data Connectors eliminated the need to use the classic "Create Bindings" (found on Data Tab -> Tables View) for CSV and Database bindings. Models created with classic CSV and Database bindings will still use the classic binding for binding to external data. Models created in 203 or later can show the deprecated table bindings if the setting *Show Obsolete Table Bindings* is set to 'True'. This property is located by going to File -> Settings -> Add-ins.
Deprecation of Wonderware Add-In
We also deprecated the Wonderware Add-In that was found on the Planning Tab -> Tables view. This deprecation is due to the addition of the AVEVA Importer and Exporter in Sprint 202.
Models that utilize the Wonderware Add-In will still use that functionality. Models created in 203 or later can show the deprecated add-in if the setting *Show Obsolete Design Add-ins* is set to 'True'. This property is located in File -> Settings -> Add-ins.

Simio Release 11 – Sprint 202 – February 27, 2020

Welcome to Sprint 202! In this sprint, we have rethought of the way Simio binds table data to external data with a new Data Connectors view. We have also added the ability to have a Simio model open with a PDF file, RandomRow and RandomValue Functionality for State Columns, and enhancements to our Camera Tracking. We finished up this sprint with improvements to our model protection capabilities and the ability to define a Primary Key in an Output Table.

Ability to Open a Model with a PDF File

We have added an option for users to open a PDF file upon opening a Simio model, similar to those PDF files automatically opened with our Examples and SimBits. This model property is located in the Project Properties panel, specifically the *Show Documentation On Load* property. To enable this feature, set this value to 'True'. The PDF file must be saved in the same directory as the Simio model and must have the same file name.



RandomRow and RandomValue Functionality for State Columns

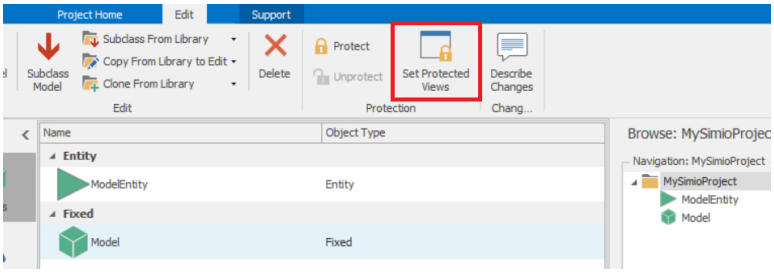
We have added RandomRow table functionality to State Columns in both Data Tables and Output Tables. One use of an Output Table is as a table that is made during the run that can be used for other modeling purposes during the same run. Your modeling needs may require you to sample from this table, for that you can use RandomRow of State Columns. The expression would be in the form of 'OutputTableName.PropertyColumnName.RandomRow'.

Camera Tracking Enhancements

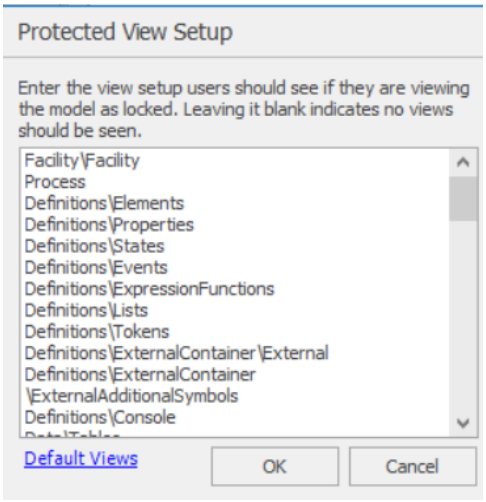
This new feature allows users to set the distance that the camera follows behind an object. This allows better animation of a model since the camera distance is now adjustable. The *Follow Distance Ratio* property is located in the Camera Tracking Options. Note that this value is a ratio so a value of '0.5' would follow the object twice as close as the default value of '1' and a value of '2' would follow the object twice as far as the default value of '1'.

Enhanced Protection Capabilities for Models – RPS and Professional Only

It has always been possible to protect a Simio model by assigning it a password. However, with this enhancement, users will now have the option to protect specific views of a model. This feature is found via the Project Folder in the Navigation Pane. Any model that is opened in Personal Edition will require entering the password upon opening the Simio model.

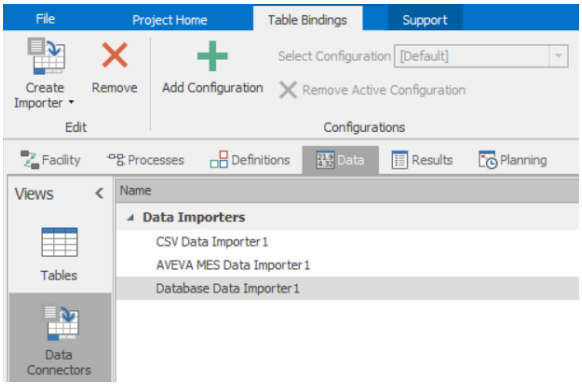


Once here, select the model you would like to protect views in and then select “Set Protected Views”. The views that you list in the Protected View Setup dialog box will be the views that will be seen once the model is protected. Leaving this dialog box empty will lead to no views being shown. Note: View syntax can be seen for all views by clicking on “Default Views”.



Data Connectors – Importers

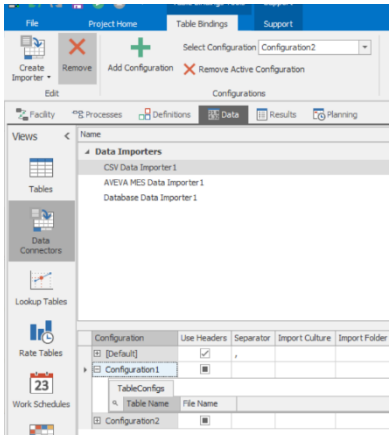
In the Data view, under the Tables view icon, there is a new Data Connectors view.

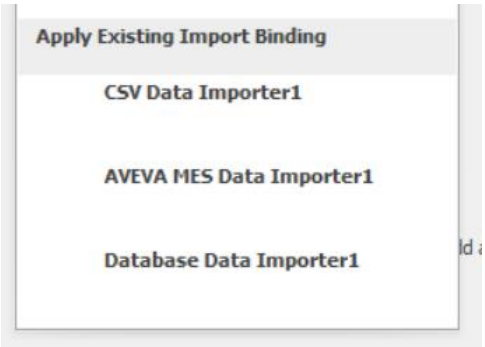


This Data Connectors view was designed to give as much flexibility as possible to the user to put configuration at the level where it makes the most sense. When a user opts to use a new connector, the data-grid will be loaded with a hierarchical master-detail view of configuration settings, an example can be seen below.

Configuration	Binding Setting 1	Binding Setting 2	Binding Setting 3
[-] [Default]		45	“fred”
Table Name	**Table Setting 1**	**Table Setting 2**	**Table Setting 3**
[-] MyTableA	45	“A”	March 12, 1980
Column Name	**Column Setting 1**	**Column Setting 2**	
MyColA	“fred”	Material345	
MyColB	“George”	Material672	
[+] MyTableB	6	“C”	April 13, 1967
[+] Testing	SomeTest		
[+] Production	SomeProd		

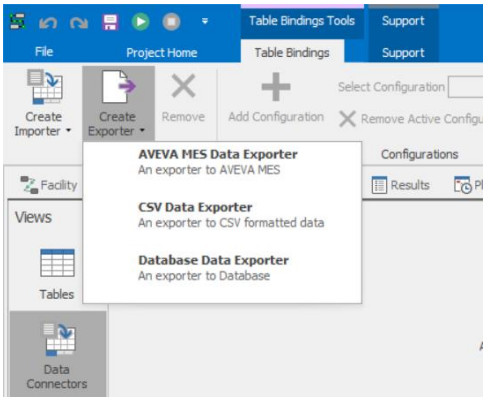
Users can specify the Import Folder where the data files are located as well as the file names to be associated with specified tables. This allows for easy changes to be made to what data file is associated to a table. Once the structure of the bindings and tables have been put in place, it is possible to Create Binding to a table binding specified in the Data Connectors. Simio currently supports a CSV Data Importer, AVEVA MES Data Importer, and Database Data Importer.

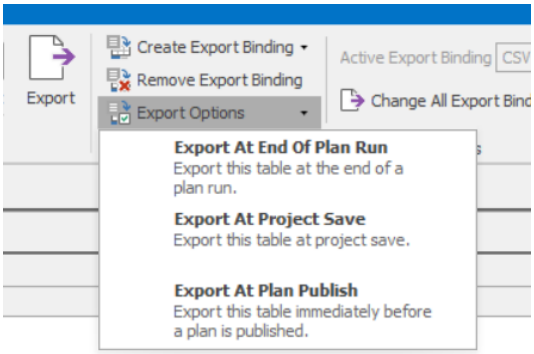




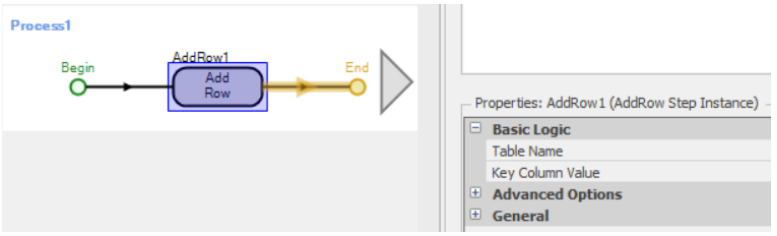
Data Connectors – Exporters – RPS Only

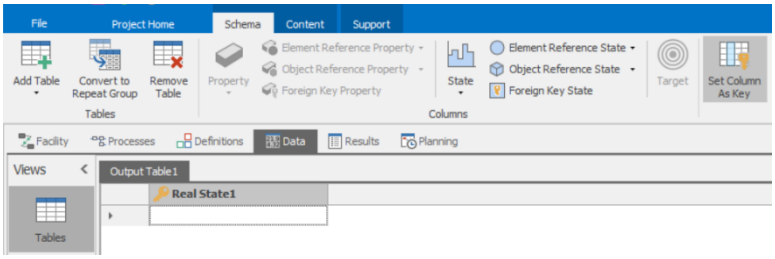
The counterpart of the newly added Importers found via the Data Connectors view is the Exporters. The Exporters allows for a binding to be created for exporting purposes on a data table. This exportation can occur at three different time: at end of plan run, at project save, or at plan publish. Simio currently supports a CSV Data Exporter, AVEVA MES Data Exporter, and Database Data Exporter.





Ability to Define a Primary Key to an Output Table – RPS Only
In this sprint, we have added the ability to set a State Column in an Output Table as a Key. Output Tables are often used to parse raw data, which can lead to a schedule being generated based on an Output Table. Due to this addition, the Add Row step now has an new property *Key Column Value* where the unique key value for the row can be placed. This feature was designed to allow child Output Table rows to be easily associated to parent Output Table rows as well as for optimizing the searching of Output Tables by providing the ability to find a row number based on the the RowForKey(key) function.

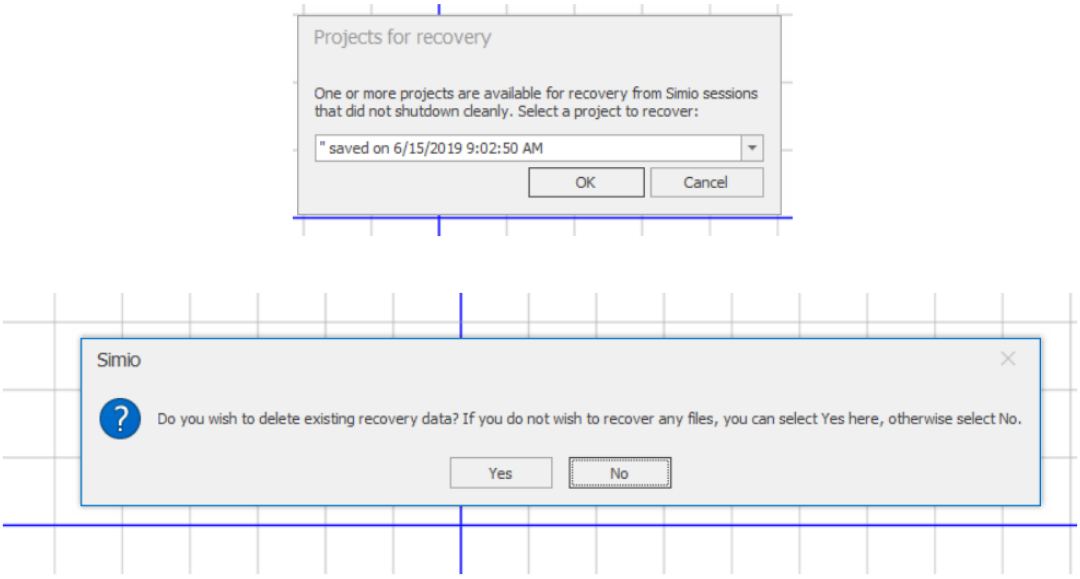




Simio Release 11 – Sprint 200/201 – February 5, 2020

Welcome to Sprint 200/201! In this sprint, we have enhanced our support for recovery project data to include an option to delete the existing auto-saved recovery file. We have also made a slight change to how multi-capacity server utilization statistics are calculated.

Project Recovery Enhancement
We have added an option for users to delete the existing data recovery file when the project is not opened for recovery. For example, if File > Settings was selected and the user specified the Project Recovery settings to auto-save, then when opening Simio, the 'Projects for recovery' dialog will prompt to open project to recover. In this sprint, if the user selects Cancel (don't open recovery file), an additional dialog prompts for the deletion of the project recovery data.



Multi-Capacity Server Utilization Statistics Change
This change was made based on customer feedback regarding utilization of multi-capacity servers.

Suppose a Server has a capacity of 2 and at time 0.0 during the simulation run, two entities simultaneously arrive to the Server and begin processing. One of the entities has a processing time of 2 hours and the other entity has a processing time of 4 hours. The Server has no output buffer and the downstream process is blocked. Thus, when an entity finishes processing, it is held indefinitely in the processing station of the Server trying to transfer out. We run the simulation for 10.0 hours.

- Previous Utilization Calculation - Server's Scheduled Utilization
  For the above example, running the model for 10 hours, we would see the Capacity.Utilized of the Server equal to 2 for the first 4 hours. Then once BOTH entities were finished processing at time 4.0 hours, the Server's ResourceState would be set to Blocked which would set the Capacity.Utilized equal to 0 for the remaining 6 hours of the run.

  The reported utilization results:
  - Server1.Capacity.Utilized.Average = (2*4)/10 = 0.8
  - Server1.Capacity.Average = 2
  - Server1.Capacity.ScheduledUtilization = (0.8/2)*100 = 40%

- New Utilization Calculation - Server's Scheduled Utilization
  For the above example, running the model for 10 hours, we would see the Capacity.Utilized of the Server equal to 2 for the first 2 hours. Then one of the entities will finish processing and we'd see the Capacity.Utilized of the Server equal to 1 for the next 2 hours. Then once BOTH entities were finished processing at time 4.0 hours, the Server's ResourceState would be set to Blocked which would set the Capacity.Utilized equal to 0 for the remaining 6 hours of the run.

  The reported utilization results:
  - Server1.Capacity.Utilized.Average = (2*2 + 1*2)/10 = 0.6

- Server1.Capacity.Average = 2
- Server1.Capacity.ScheduledUtilization = (0.6/2)*100 = 30%

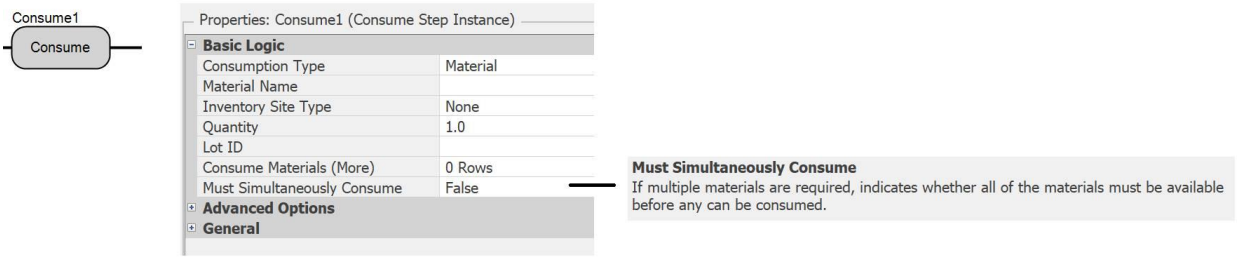
Note that this change only impacted the utilization statistics reported for multi-capacity Servers. For models saved in prior Sprint versions, to opt in to the new utilization calculation approach, the user must go to Run Setup -> Advanced Compatibility Settings in the Run Ribbon and set the compatibility setting for *Exclude Entities Transferring Out Of Child Stations From Resource Utilization* to 'True'.

Simio Release 11 – Sprint 198/199 – December 13, 2019

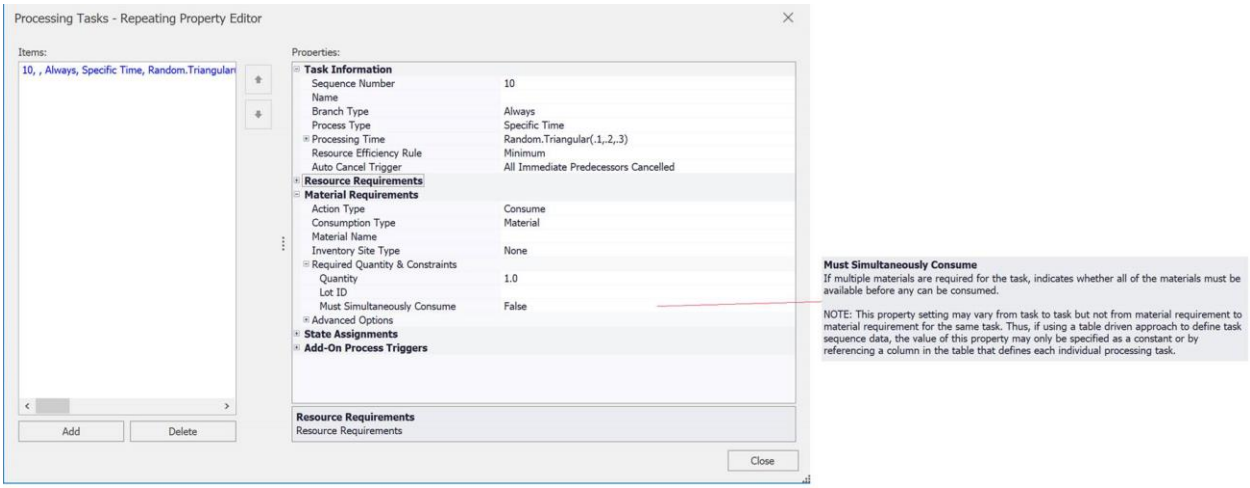
Within this sprint, we have added several user-requested features, including simultaneous material consumption, improved search table row referencing, multi-pass simulation enhancements for state columns/output tables and enhanced search capability within RPS Gantts.

Material Consumption Enhancements
We have enhanced several material consumption areas of the software, including the Consume step and related material consumption within the processing tasks of Server, Combiner and Separator objects. These were user-requested features.

The Consume step now has a new *Must Simultaneously Consume* option as illustrated below:



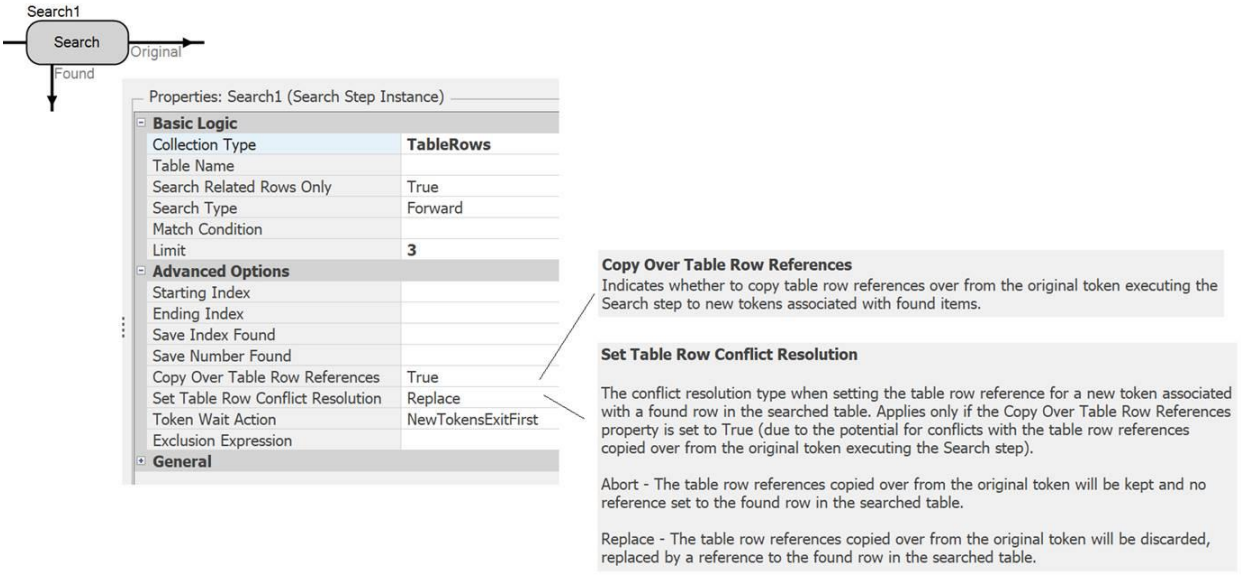
A *Must Simultaneously Consume* option is now provided, under Processing Tasks -> Material Requirements -> Required Quantity & Constraints, for material consumption within the Server, Combiner and Separator objects.



Note that the Material Requirements property categorization now has a *Required Quantity & Constraints* category like Resource Requirements.

Search Step Behavior - Enhancements
The Search now has two new properties in Advanced Options (see below). A *Copy Over Table Row References* property that applies when searching any collection type, and a *Set Table Row Conflict Resolution* property that applies only if searching table rows.

Note that the default value for the *Set Table Row Conflict Resolution* property is 'Replace' for new models, but 'Abort' for old models (as that was the implicit behavior in previous Simio versions).



There is also a new Advanced Compatibility bit which enables some recommended Search step behaviors, eliminating flaws in the previous searching process.

For new models, this compatibility setting will be True, but False for older models unless manually changed.

Search Step Uses Explicit Copy Over Table Row References
Indicates whether the copying of table row references over from an original token executing a Search step to new tokens associated with found items is based solely on the step's Copy Over Table Row References property value, removing all restrictions that were required by previous Simio versions. Enabling this option also ensures that the table row references for the original token are always included in match condition and search expression evaluations. The recommended value for this setting is True.

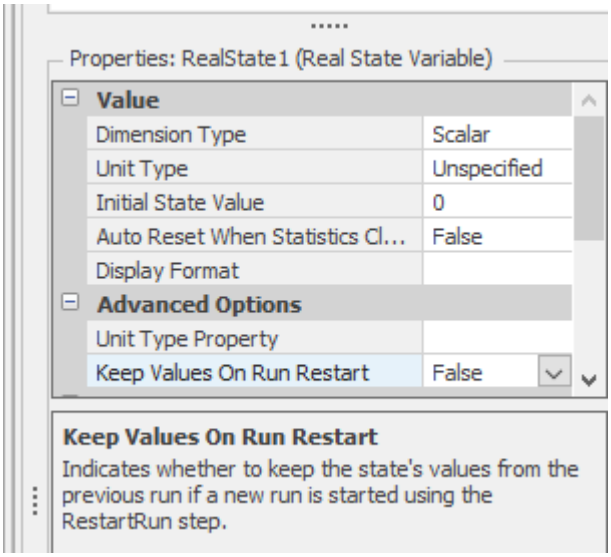
The Search step is a popular step that is heavily used in consulting projects, often to search rows in tables. When searches of table rows are performed, there may be table row references set on the original token, that the user wants included in match condition or search expression evaluations and/or copied over to new tokens associated with found rows in the searched table.

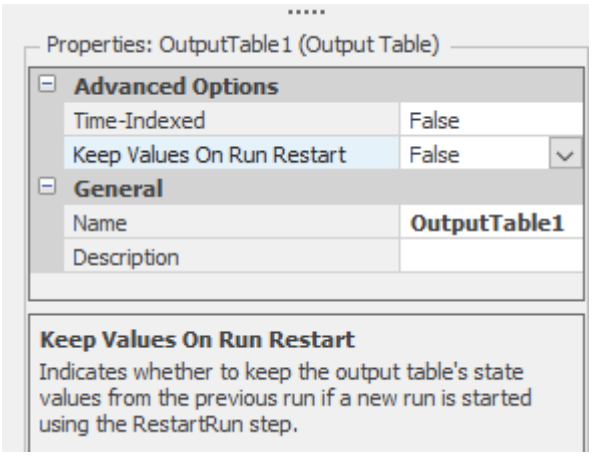
Depending on the table relationship design, there can be conflicts between the table row references set on the original token and the internal 'evaluator' token used by the Search step to evaluate each individual row in the table, or a conflict when trying to set the table reference for a new token to the found row.

Table State Columns and Output Tables Enhancement for Multi-Pass Simulation Approach - RPS Only
In Sprint 197, we added a new RestartRun step and Run.RunNumber function. We have now added the ability specify a *Keep Values on Run Restart* property for both Data Table state columns (ones that are not element or object reference states) and Output Tables. The options are only be available for RPS licenses. These features work across interactive, plan, and experiment runs.

For Table State Columns:

For Output Tables:





Note that State Columns in Output Tables DO NOT have the option individually, only state columns on standard Data Tables do. Output Tables either keep the entire table or do not.

These features support a multi-pass simulation approach as described below:

Single-Pass vs Multi-Pass Simulation Approach

Typically, creating a resource schedule or generating the results for a scenario replication instance (experiment mode) requires only one run of the simulation model. A model run is initialized, is executed for some run length, and then ended, completing the Create Plan or Run a Replication action.

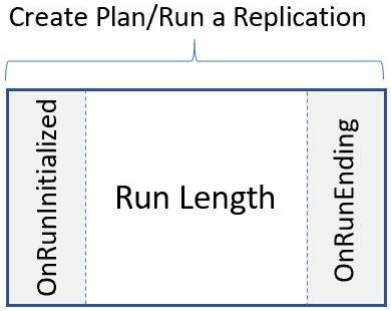


Figure 1 - Single-Pass Simulation Approach

There are cases however where it is insufficient to run the model only once to create a schedule or generate results for a scenario replication instance. Instead, the model must be run multiple times in succession, with information determined by previous runs used as input data for the next run. Simulation runs are continued in succession until the desired objective is achieved. This type of simulation approach is often referred to as a *multi-pass* approach.

Page Break

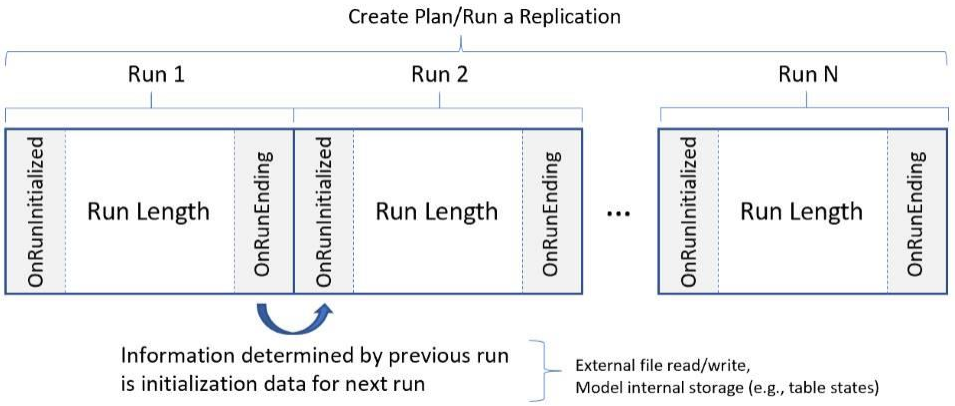
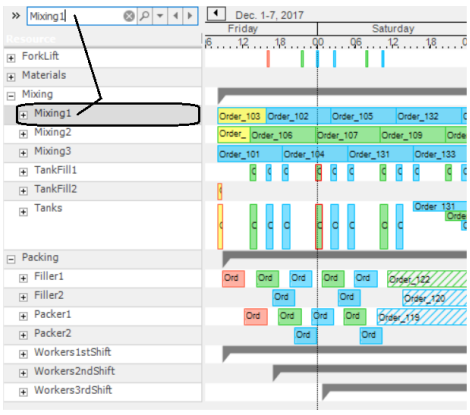


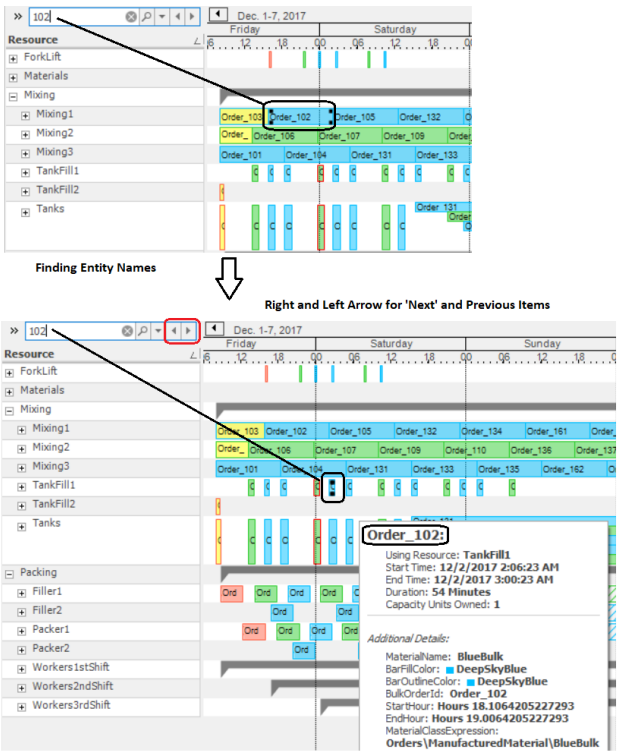
Figure 2 - Multi-Pass Simulation Approach

Continuous Search Capability within Gantts - RPS Only

We have enhanced the Search capability within both the Resource Plan and Entity Workflow Gantts to allow the user to perform continued searches throughout the Gantts. The new down arrow shows a list of previously searched items. The right and left arrows allow for forward and backwards searches through the Gantts on a row by row basis.

For example, in the screenshot below (right side), when the user searches for '102', the first instance of '102' appears in Order_102 within the Gantt timeline area in the Mixing1 row. Selecting the right arrow will then move the highlight to the next '102' instance in the TankFill1 row. Searched items will be highlighted on a row by row basis, without respect of timeline. Thus, multiple instances on a single row will be highlighted before instances on a lower row in the Gantt.



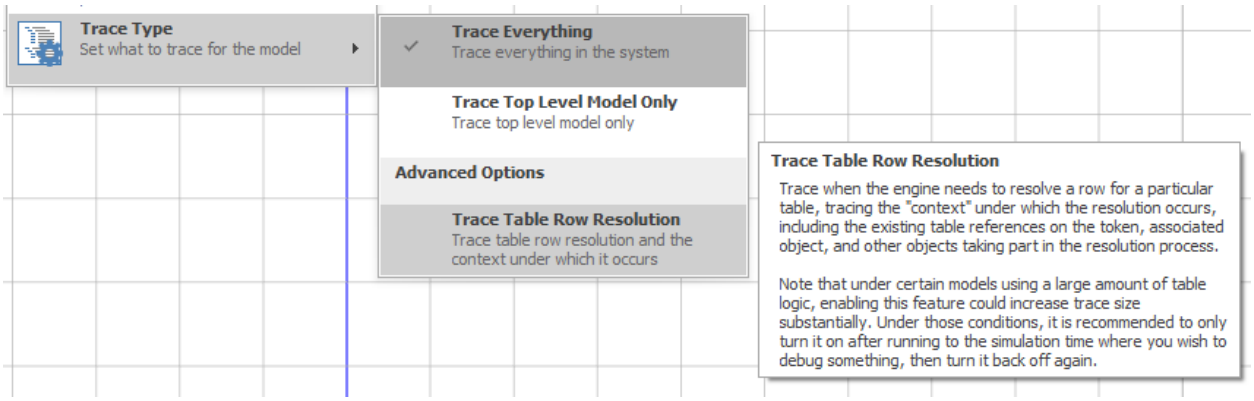


Searches are NOT case-sensitive. Searches include items in the Constraints or Tasks and will expand the + on a row to show the Constraint / Task item if not already expanded. Searches do not include items listed in the details tooltip (opened when hovering over an area on the Gantt).

Simio Release 11 – Sprint 196/197 – October 10, 2019

Within this sprint, we have enhanced our help for table referencing, added documentation for scheduling labor efficiencies example and updated our examples animations.

Trace Enhancement - "Trace Table Row Resolution" advanced debug/tracing
There is a new tracing option (off by default) to trace additional information whenever the runtime is attempting to resolve a table row:



In the trace below, the feature is turned on. This can be helpful, as a user may realize “My tbl_Routings reference went from [11] to [12] (sequence table), and any foreign keys pointing into that row might have changed to (foreign key state in an output table). Maybe that is why I’m getting this odd runtime error.”

[Begin]	Process 'Output@serv_Product_Hoppers2.OnEnteredFromAssociatedObject' execution started.
[Assign] OnEnteringAssignm...	Resolving row for table 'tbl_ManufacturingOrders'...
	Associated object (ent_FGOrder.124) references:
	Table references: tbl_Materials[31], tbl_ItemMaster[0], tbl_Routings[11], tbl_ManufacturingOrdersOutput[30], tbl_Manufacturin...
	Explicit selection: tbl_Materials[31], tbl_ItemMaster[0], tbl_Routings[11], tbl_ManufacturingOrdersOutput[4], tbl_ReworkWH[0]
	Potential rows: tbl_MaterialLots, tbl_FermOrders, tbl_BioConvOrders, tbl_BillOfMaterials, tbl_ManufacturingOrdersOutput
	Assigning state variable 'Model.v_TestString' the value '27_43018'. Previous value was 'a'.
[Fire] EnteredEvent	Firing event 'Output@serv_Product_Hoppers2.Entered'.
[Decide] IfTransporter	Token branching on condition 'Entity.Is.Transporter'.
	Token sent to 'False' exit.
[Execute] RoutingOutLogic	Executing process 'Output@serv_Product_Hoppers2.RoutingOutLogic'.
[Begin]	Process 'Output@serv_Product_Hoppers2.RoutingOutLogic' execution started.
[SetNode] NextDestination	Destination node of entity 'ent_FGOrder.124' set to value 'Input@serv_PackLine1'.
[End]	Process 'Output@serv_Product_Hoppers2.RoutingOutLogic' execution ended.
[Transfer] ToOutboundLink	Entity 'ent_FGOrder.124' transferring from '[Node] Output@serv_Product_Hoppers2' into node 'Input@serv_PackLine1'.
[Begin]	Process 'Input@serv_PackLine1.OnEnteredToAssociatedObject' execution started.
[Assign] OnEnteringAssignm...	Resolving row for table 'tbl_ManufacturingOrders'...
	Associated object (ent_FGOrder.124) references:
	Table references: tbl_Materials[31], tbl_ItemMaster[0], tbl_Routings[12], tbl_ReworkWH[0]
	Explicit selection: tbl_Materials[31], tbl_ItemMaster[0], tbl_Routings[12], tbl_ReworkWH[0]
	Potential rows: tbl_MaterialLots, tbl_FermOrders, tbl_BioConvOrders, tbl_ManufacturingOrdersOutput, tbl_BillOfMaterials
	Unable to get value of property 'Input@serv_PackLine1.AssignmentsOnEntering.AssignmentsOnEnteringNewValue'.

New RestartRun Step and Run.RunNumber Function – RPS Edition Only

We have added a new RestartRun step may be used to set the ending time of the simulation run to the current time. This will cause the run to end once all simulation events scheduled for the current time have been processed. Interactive mode will require the next run to be started manually. Otherwise, a new run will be automatically started, treated as a restarted run for the same scenario replication instance if running in experiment mode.

If in interactive mode, using the Reset button to re-initialize the model to its starting conditions will cancel a restart command and set the run count using the new function, Run.RunNumber, back to 1.

We have added a new function **Run.RunNumber** that returns a one-based run count, incremented if a run restart occurs using the RestartRun step.

For example, if executing a simulation with run restarts, then for the first run you'd see Run.RunNumber return 1. For the next run, if restarted using the RestartRun step, you'd see Run.RunNumber return 2 and so on. An example of using the Run.RunNumber function in process logic might be in a condition expression to decide whether to execute a RestartRun step. For example, if implementing a two-pass algorithm, then you'd only want to execute the RestartRun step if 'Run.RunNumber==1'.

Bill of Materials Inventory Enhancement
We now allow Bill of Material (BOM) consumption to include entries for both location specific and not location specific (previously all or nothing for a BOM that is location based). This was a user-requested feature.

The *Inventory Site Type* property now reads "Indicates the fixed object that is the inventory location. Applies only to material elements whose *Location Based Inventory* property is set to True." Thus, when the inventory for a material in a model is a single location-less inventory, there is no ambiguity about the inventory reference when trying to consume, produce, or reserve that material. We relaxed the Simio engine's error checking to simply ignore the *Inventory Site Type* settings if the *Location Based Inventory* property of a material is set to 'False.'

SchedulingLaborEfficiencies Documentation and Updated Examples Animations
We have added documentation for one of our new scheduling examples that was recently added to the installation. We've also updated some of our example animations, there will be more enhancements to come in the future.

Simio Release 11 – Sprint 195 – September 19, 2019

Within this sprint, we have added the option to seize/release by condition with Secondary Resources (Other Resource Seizes and Other Resource Releases), as well as an additional option for Table Based Work Schedules to determine how a schedule start date relates to the start and end date times.

Secondary Resources Enhancement – New *Skip Seize If and Skip Release If* Properties

For completeness, we have added 'Skip Seize If' and 'Skip Release If' advanced options to the *Secondary Resources -> Other Resource Seizes/Other Resource Releases* repeat groups. These properties allow users to optionally skip the seize or release requirement.

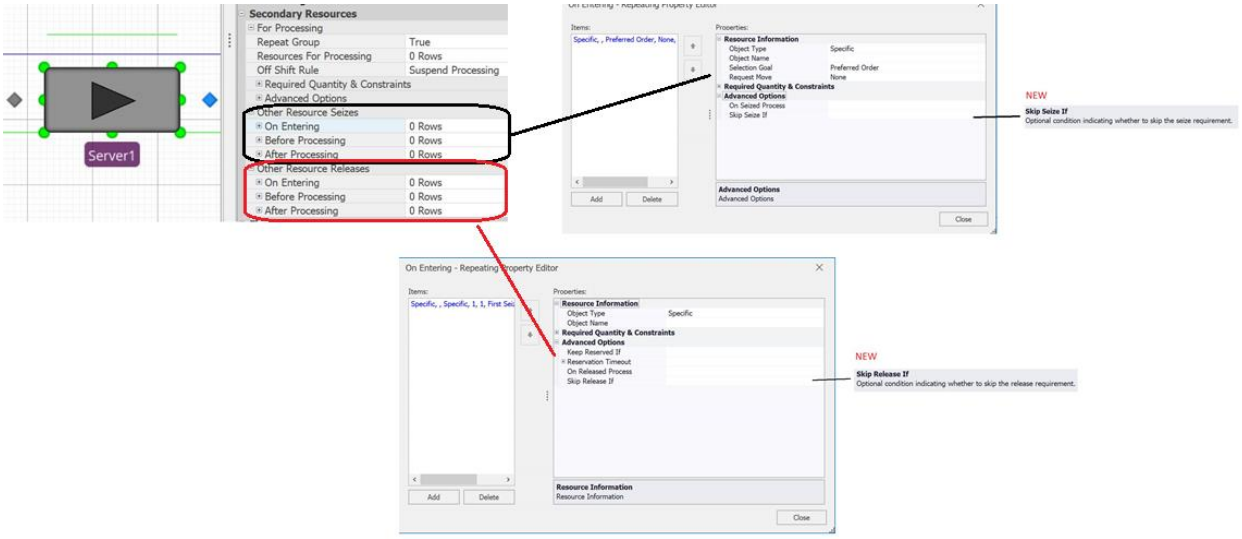
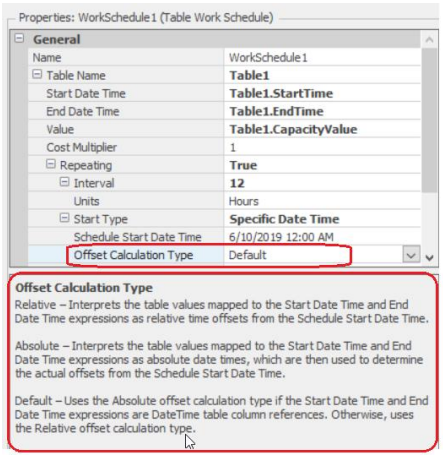


Table-Based Schedules Enhancement

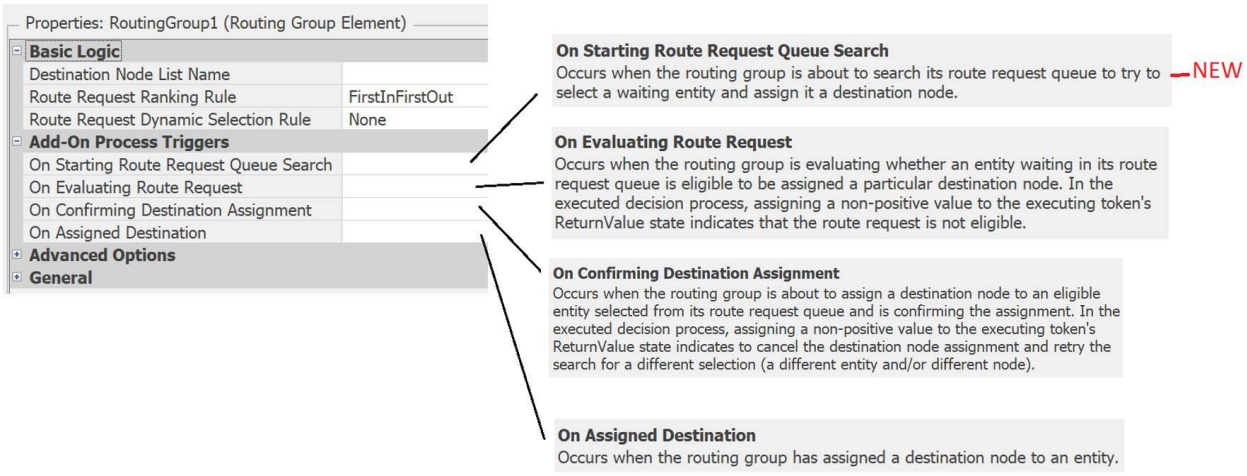
In sprint 193, we added a *Start Type* property to allow users to specify an offset starting time for table-based schedules. Then, in sprint 194, we enhanced the *Start Type* property with an option for 'Specific Date Time', where the *Schedule Start Date Time* property is then specified, as shown below. Now, in sprint 195, we've included a new *Offset Calculation Type* property when using the *Start Type* of 'Specific Date Time'. This can be set at 'Default', 'Relative' or 'Absolute' and determines how the *Schedule Start Date Time* relates to the *Start Date Time* and *End Date Time* values.



Simio Release 11 – Sprint 194 – August 29, 2019

Within this sprint, we have enhanced the RoutingGroup element, Table-based Work Schedules, and RPS logs with new properties. We have also included optional seize / release conditions within secondary resources and material requirements. Finally, breakpoints may now be set on decision type processes.

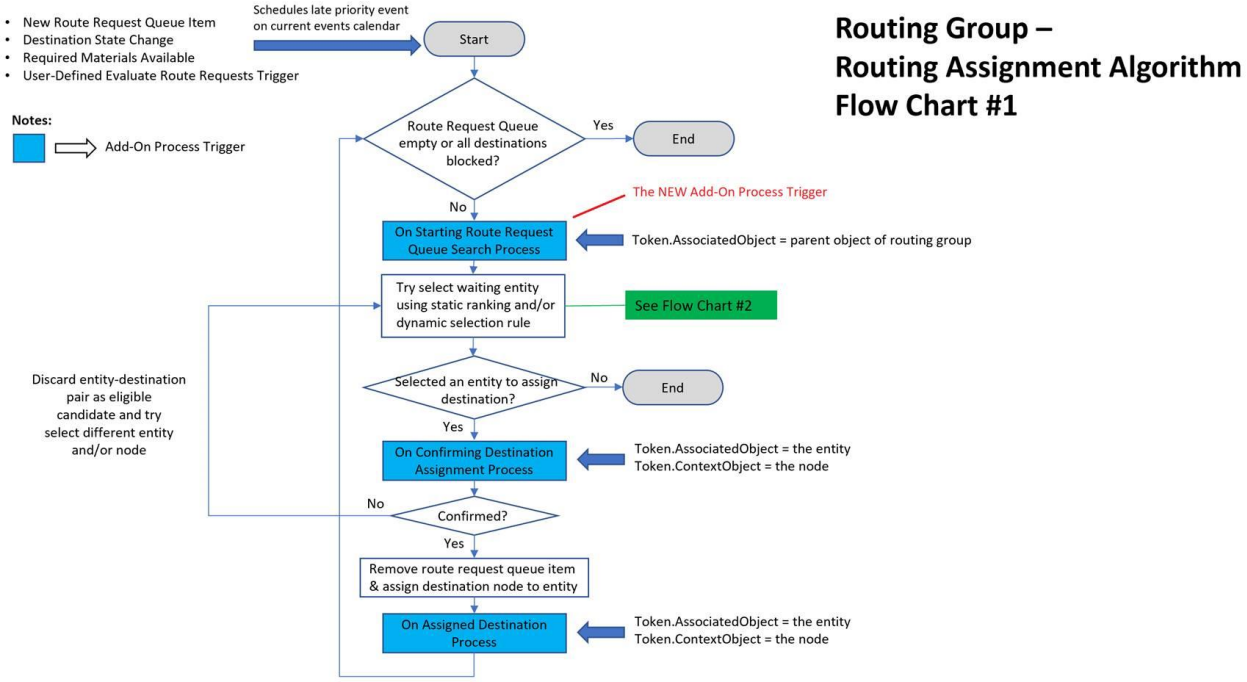
RoutingGroup Element Enhancement
The RoutingGroup element has been enhanced to provide a new *On Starting Route Request Queue Search* add-on process trigger. This was added to provide greater flexibility to the RoutingGroup as well as eliminate unnecessary process execution.



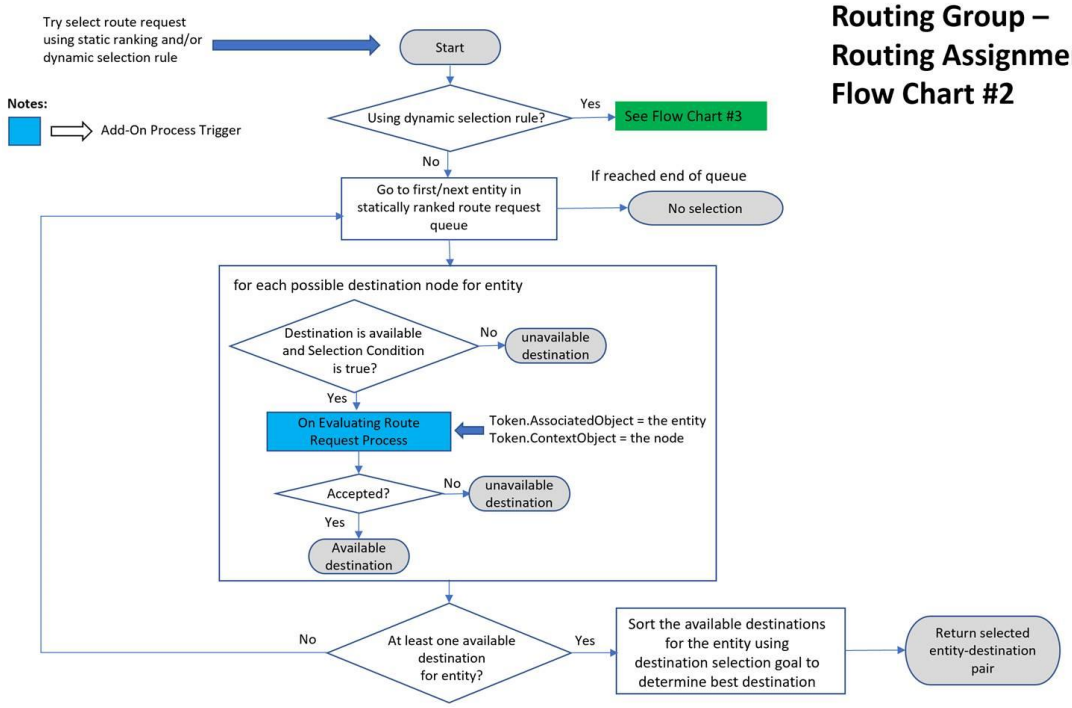
Example Scenario:

You have 20 entities waiting in the routing group's route request queue. And there are 4 available destination nodes. To assign entities to the four available nodes, you'd see the *On Starting Route Request Queue Search* process get executed 4 times (once per each search pass through the queue) and the *On Evaluating Route Request* process get executed 190 times (20 X 4 times during first search pass + 19 X 3 times during second search pass + 18 X 2 times during third search pass + 17 times during fourth search pass).

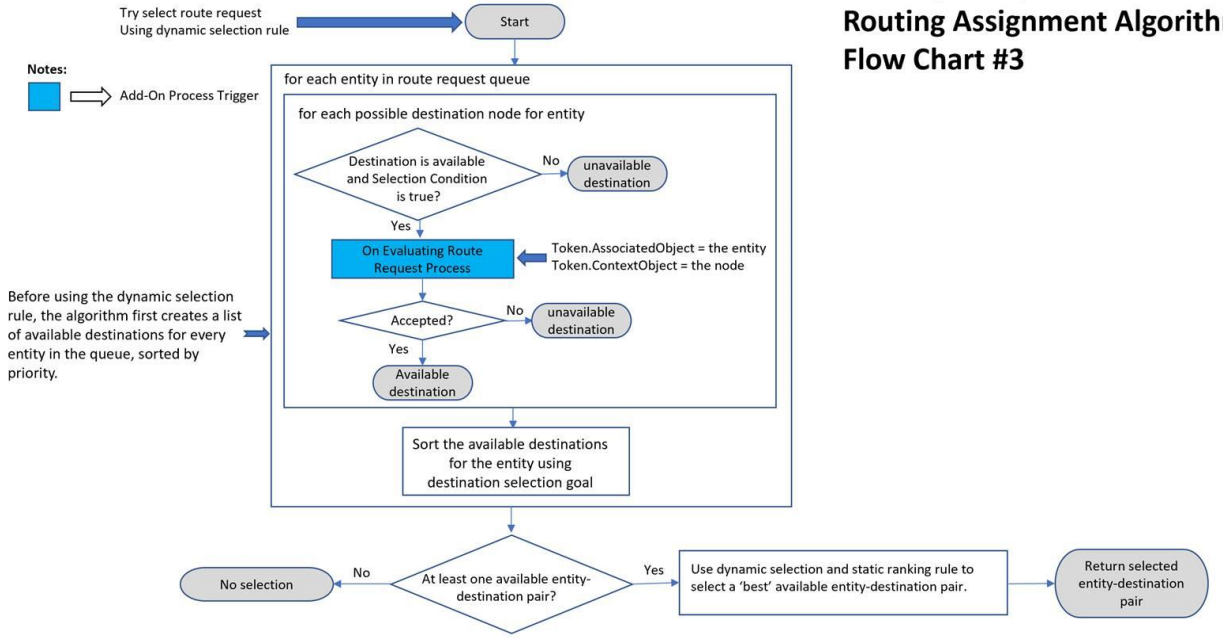
Routing Group – Routing Assignment Algorithm Flow Charts



Routing Group – Routing Assignment Algorithm Flow Chart #2



Routing Group – Routing Assignment Algorithm Flow Chart #3



New TableBindings View within Data Tab

We have added a new TableBindings view, located within the Data tab below the Tables view. This has been added to support additional "Modern" binding type features that Simio will be adding over the next few sprints. Currently, this view displays "Classic" table binding names with their data table reference, as specified in the Tables view.

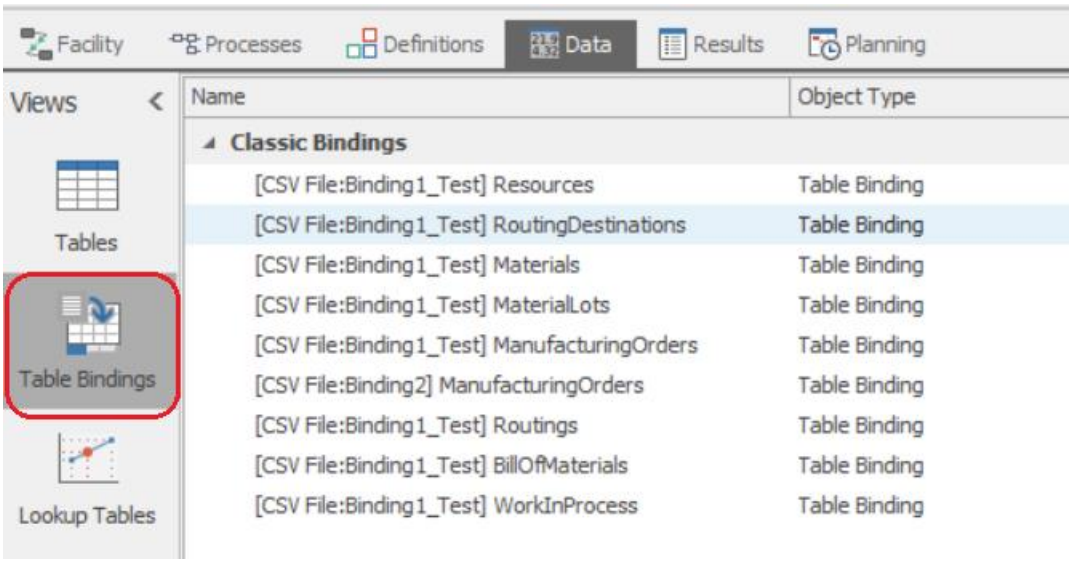
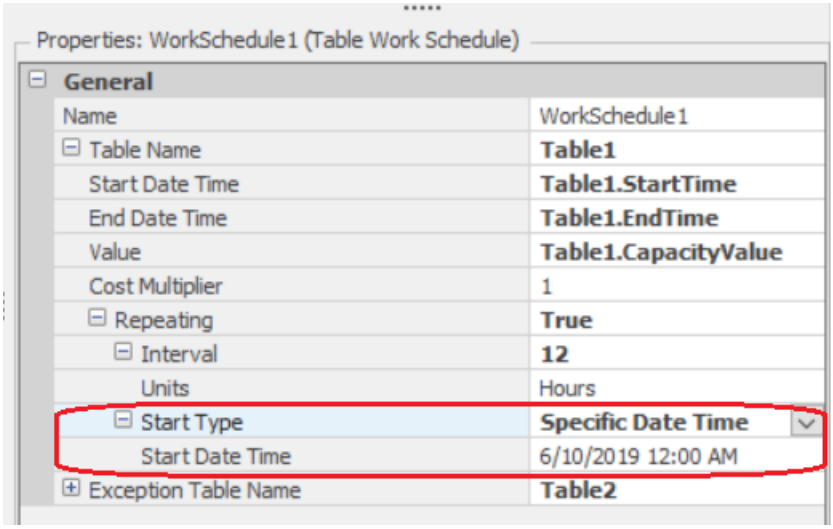


Table-Based Schedules Enhancement

In sprint 193, we added a *Start Type* property to allow users to specify an offset starting time for table-based schedules. We've now enhanced this property with an option for Specific Date Time, where the *Start Date Time* property is then specified, as shown below.



Standard Library Objects Enhancement - Change 'Setup' Resource State from a Utilized to Non-Utilized

For greater compatibility with the standard Overall Equipment Effectiveness (OEE) definition, the resource utilization calculation setting of the 'Setup' and 'OffShiftSetup' resource states of the Server, Combiner, and Separator objects has been changed so that allocated capacity while in those states is no longer considered utilized capacity.

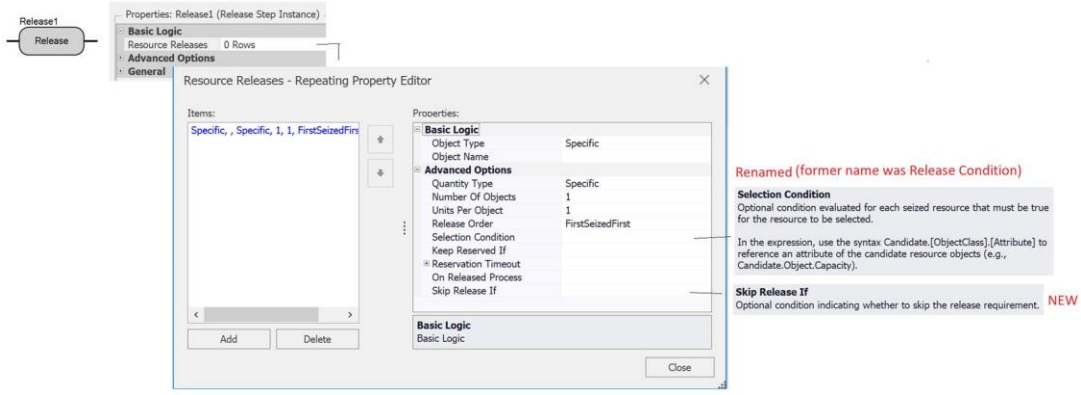
IMPORTANT NOTE: This means that, in Sprint 194 software, if you update models to the new Server/Combiner/Separator definitions and there are tasks of process type 'Sequence Dependent Setup', then you'll now see lower utilization statistics for those objects where the setups occurred. If you

have specified usage cost rates at the object, those usage cost rates will also no longer be accrued by entities during the setup times.

Release Step Enhancement - 'Skip Release If' Property

The Release step now provides a *Skip Release If* property, similar in functionality to the Seize *step's Skip Seize If* property. It allows for an optional condition that indicates whether to completely skip that particular release requirement (i.e., simply ignore the repeat group row).

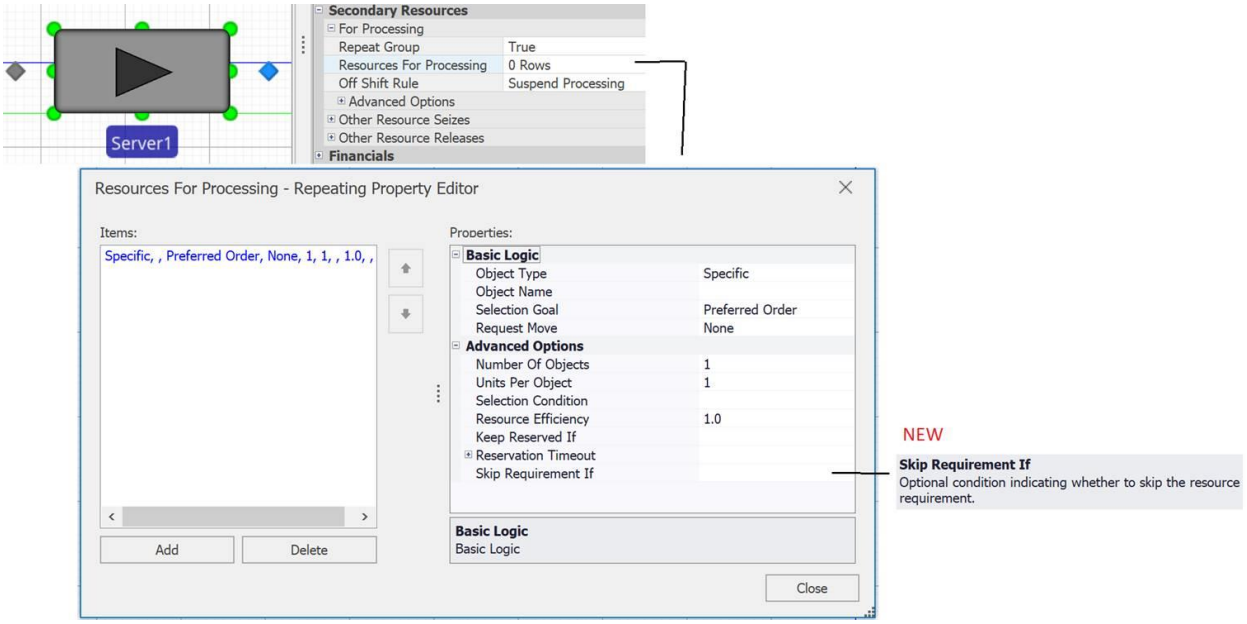
The Release step's *Release Condition* property has been renamed to *Selection Condition* to be consistent with the Seize step and to help lessen any possible confusion that might have arisen between having *Release Condition* and *Skip Release If* properties.

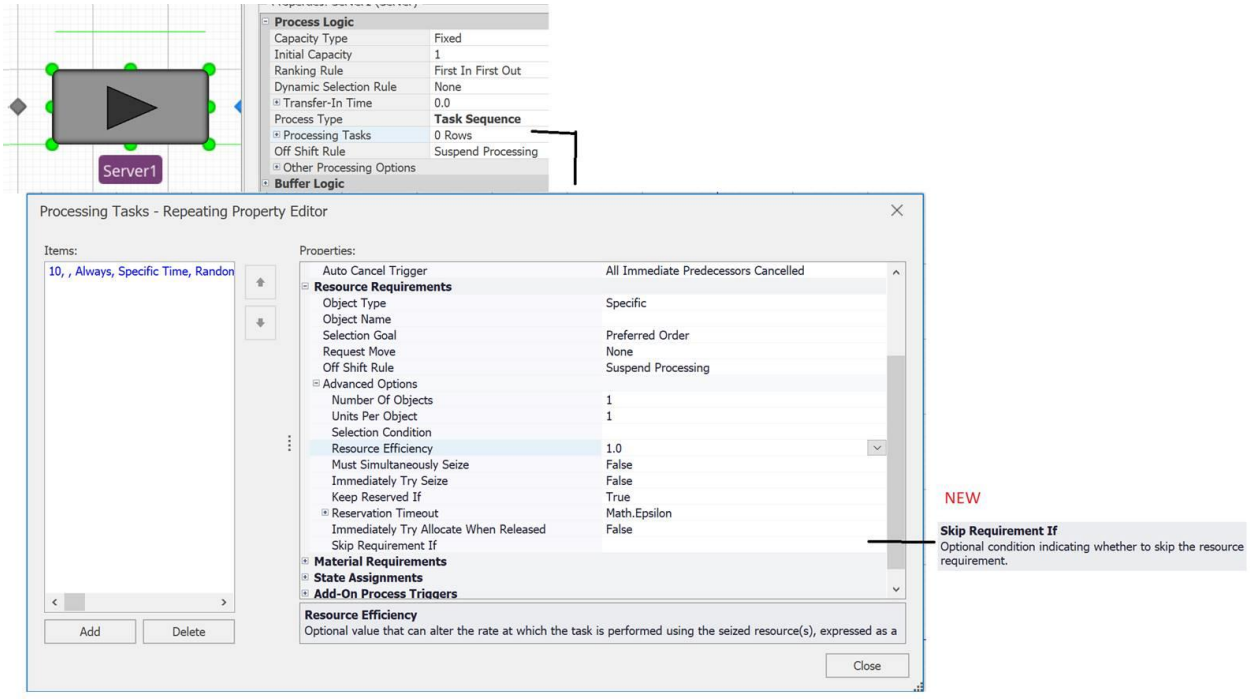


Secondary Resources Enhancement – New *Skip Requirement If* Property

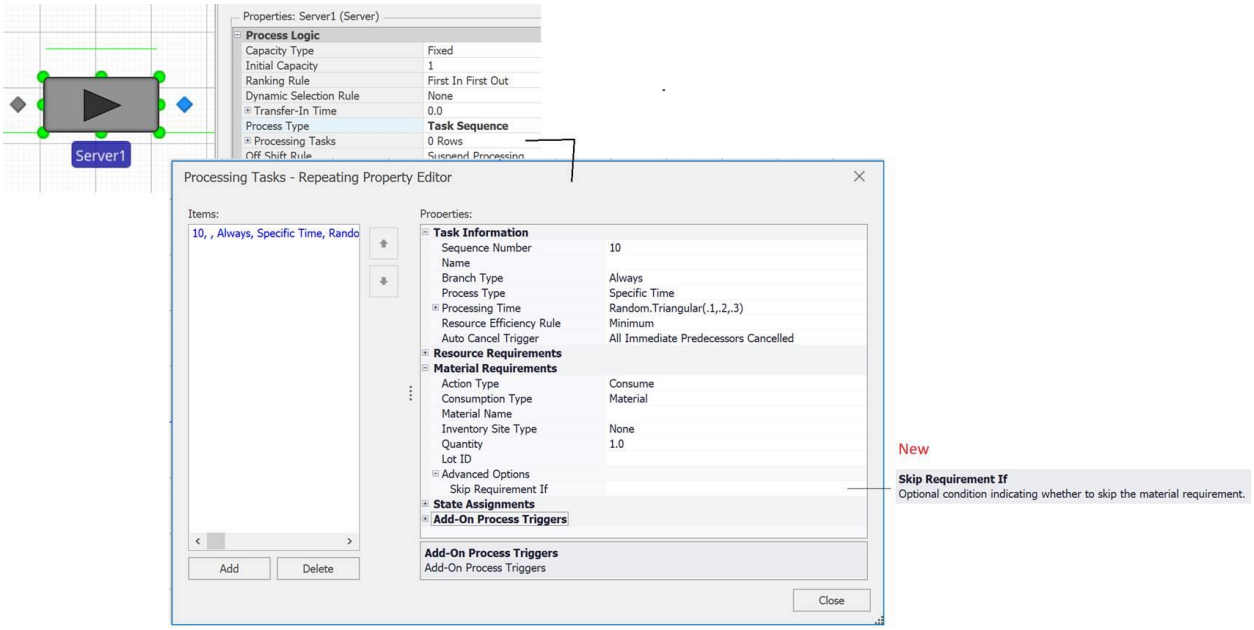
In the Server, Combiner, Separator, Filler, & Emptier, for either *Secondary Resources For Processing* requirements or *Task Resource Requirements*, a new *Skip Requirement If* expression property is now provided. This feature allows for an optional condition that indicates whether to completely skip a resource requirement (both the seize and release steps).

For example, a user may have a modeling situation where a worker is only required sometimes for a particular task, thus needing the ability to filter out resource requirements on a requirement by requirement basis without having to define different tasks. For example, suppose the *Object Name* property for a resource requirement is specified as ModelEntity.WorkerReference where WorkerReference is a state variable on the entity that is set to Nothing if no worker is actually required. You would specify the *Skip Requirement If* conditional expression for that resource requirement as 'ModelEntity.WorkerReference==Nothing'.





Processing Tasks – Material Requirements Enhancement – New *Skip Requirement If* Property
In the Server, Combiner, & Separator, for *Task Material Requirements*, a new *Skip Requirement If* expression property is now provided. Similar to the above feature for secondary resources, this feature allows for an optional condition that indicates whether to completely skip a material requirement (e.g., filter out a particular consume or produce material action for the task).



Decision Processes - Allow Breakpoints

Simio now will break on a breakpoint in a decision process, as long as you have the 'Use Scheduled Breakpoints' setting in File -> Settings set to 'False' (which is the default). Some of the decision processes within Simio include OnEvaluatingMoveRequest, OnEvaluatingRiderAtPickup, OnEvaluatingRiderReservation, and OnEvaluatingSeizeRequest.

RPS Logs Enhancement - Allow Custom Log Expression Columns in All Logs

We now support the addition of new expression columns in all logs. Previously, the user could only add extra columns to the Resource Usage, Resource Info or Task Logs.

For logs that have both a start time and an end time column, you can set the expression to evaluate at the start or end. Note that even though the Transporter Usage Log has four datetime columns, there's no provision for evaluating an expression at either Start Ride Time or End Ride Time.

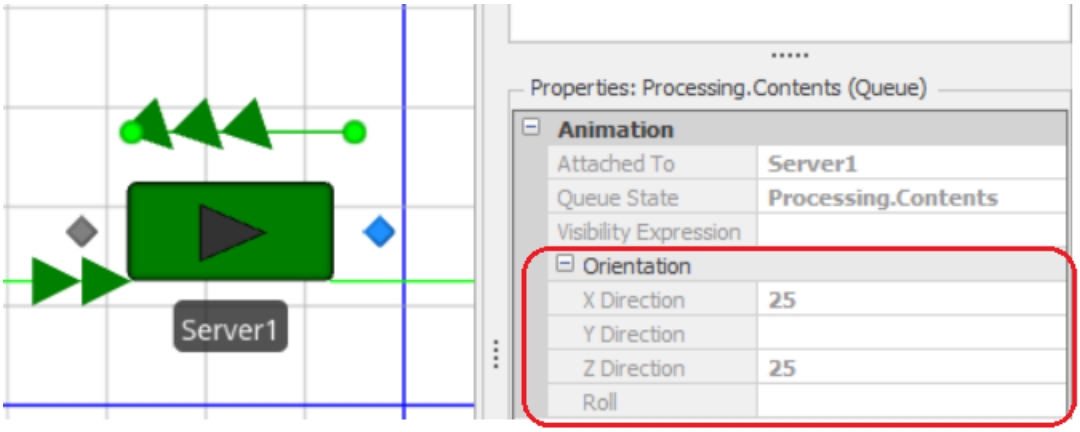
Additional columns are useful in various logs for developing dashboards and table reports as well as filtering data.

Simio Release 11 – Sprint 193 – August 8, 2019

Within this sprint, we have added several user-requested features, including entity orientation options within queues, Material element replenishment policy properties and repeatable dynamic selection rule options.

Gantt Drag and Drop Enhancement
We have added optional *Routing Group* for Node stipulations, and optional *Seize Location* for Resource stipulations for the Stipulate step. These are both logged to the Resource Usage Log, and that information is used for drag-n-drop on the Gantt. This feature was added to sprint 192+ and documented in this sprint.

Entity Animation Enhancements
We have added ability to specify an optional Direction X/Y/Z and Roll for orientation of entities on queue animations. This was a user-requested feature.



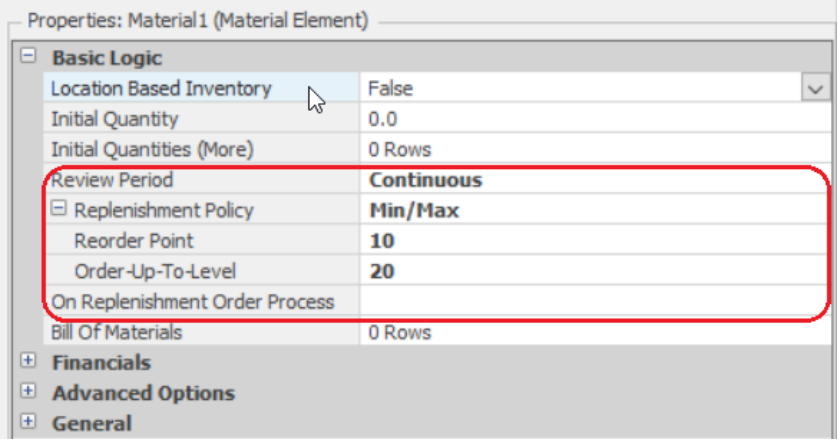
This orientation takes precedence over any other given orientation for the queue (for example, the direction specified by an "Oriented Point" queue).

Additionally, there are some new functions in the Orientation namespace for objects:

Orientation.Direction.X – Returns the X component of the unit vector indicating the current direction the object is facing.
Orientation.Direction.Y – Returns the Y component of the unit vector indicating the current direction the object is facing.
Orientation.Direction.Z – Returns the Z component of the unit vector indicating the current direction the object is facing.
Orientation.Roll – Returns the current roll angle in degrees, measured clockwise from the +Y, of the object in the X-Y plane.

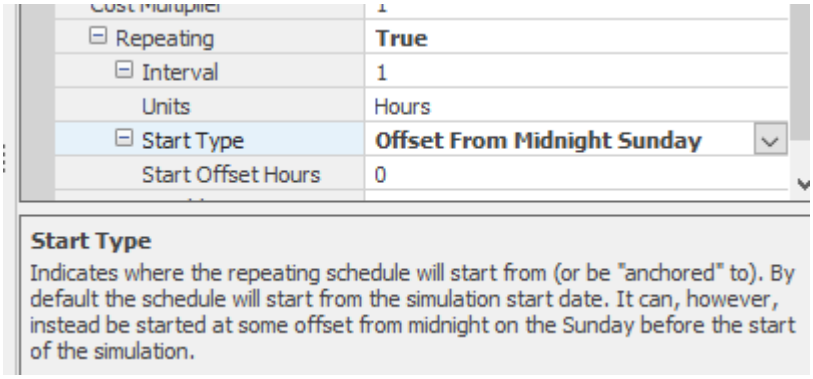
Materials - Review Period & Replenishment Policy If Not Location Based Inventory
If the *Location Based Inventory* of a Material element is set to 'False', we now allow *Review Period* and *Replenishment Policy* related properties to be specified for the single global inventory of that material.

This information used to only be specified with the Inventory element for the material. This was a user-requested feature from our Simio Sync user's meeting.



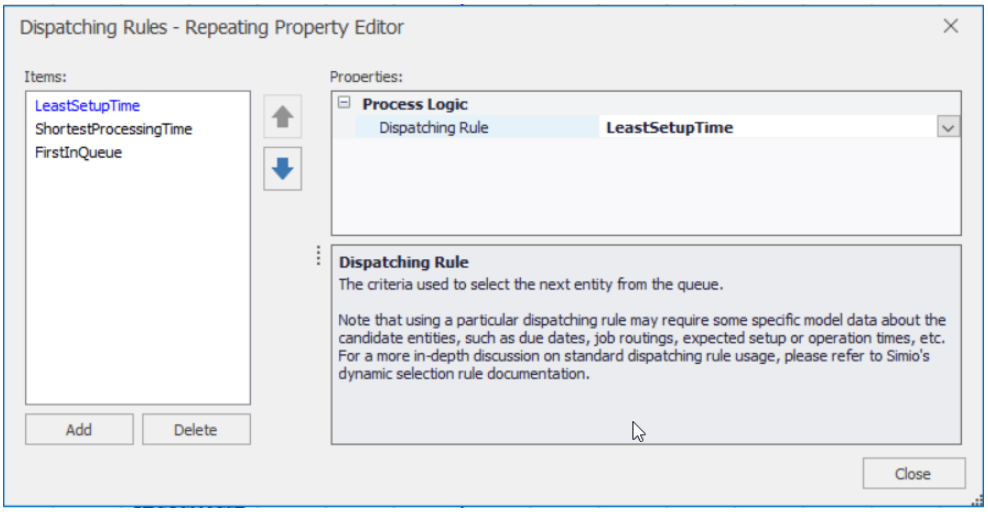
Materials - Consuming/Producing Location Based Inventory - Support 'Lot ID' Property
We have added the ability to specify the lot identifier property, *Lot ID*, on location- based inventory within the Consume and Produce steps. This then also allows the *Lot ID* within the Task Sequences Material Requirements section of properties when *Inventory Site Type* is specified. This was also a user-requested feature.

Table Based Schedules - Change "starting type" for repeating table-based schedules
Users can now give an alternate starting type/anchor point for a repeating table schedule. By default, it starts the schedule from the start of the run. Now users can alternatively tell it to start from some offset from midnight on the first Sunday before the run. If a repeating schedule is raw numbers for offsets (instead of datetimes), in terms of week- long repeating intervals, this mode is probably more appropriate. This is an enhancement to the recently added repeating schedules.



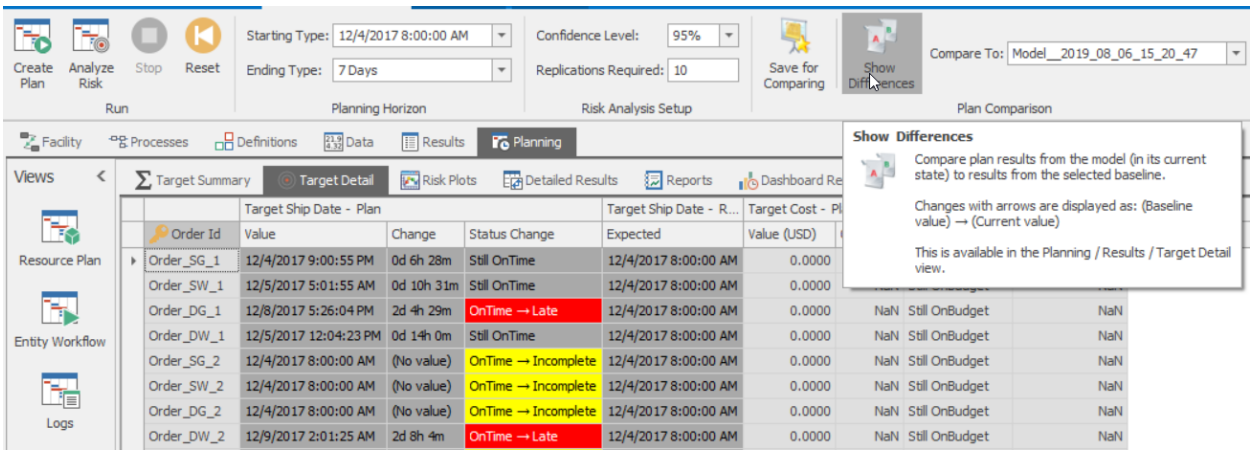
Dynamic Selection Rule – Enhancement
We have added a *Dispatching Rules* repeat group to support multi-level tie breaking within the *Dynamic Selection Rule* property. Users can keep the current configuration (*Repeat Group* property is 'False') where *Dispatching Rule* and *Tie Breaker Rule* are visible. Alternatively, users can use the new option (*Repeat Group* property is 'True'), where multiple levels of the *Dispatching Rule* property can be specified.

This enhancement has been made in all areas of the product that use Dynamic Selection Rules. This includes Server, Separator, TransferNode, Worker and Vehicle objects in the Standard Library, Emptier and Filler in the Flow Library and RoutingGroup and Station elements.



RPS Enhancement - Compare Plans without running Risk Analysis

We now allow the user to compare plans without running risk analysis, simply using Create Plan button. The Save for Comparing button is now available after using the Create Plan to generate the deterministic plan. This saves a copy of the model in the Baseline Folder in the Navigation Window. After a run was saved for comparing, changes can be made to model properties and/or data tables and Create Plan button used to run new plan. Under Results > Target Detail report, select the show Differences button (as seen below) to show the status change between the plans.



Simio Release 11 – Sprint 192 – July 18, 2019

Within this sprint, we have added the capability to model resource efficiencies, as well as repeatable secondary resources for processing. The ability to add or delete a node from an existing link is a much-requested feature that's been added as well.

Resource Efficiencies

We have added a number of features, detailed below, that will allow an *efficiency* value, expressed as a fraction, to be associated with any seized resource – particularly a secondary resource seized for processing or task processes within a Server, Combiner, or Separator object. The actual work duration is the planned work duration divided by the resource efficiency. Typically, the efficiency would be defaulted to 1.0. Values greater than 1 shorten the time and values less than 1 lengthen the time.

For example, if a task takes one hour at a 100% resource efficiency rate, then

If Resource Efficiency Is	**Then Task Duration Will Be**
1.5 (150%)	0.667 Hours
1.0 (100%)	1.0 Hours
0.8 (80%)	1.25 Hours
0.5 (50%)	2.0 Hours

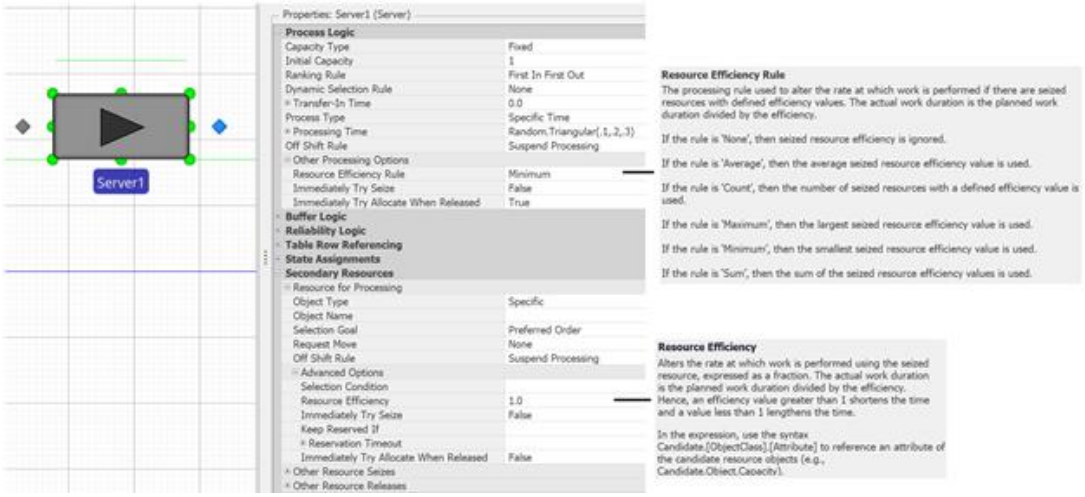
If there is more than one seized resource with an efficiency, there is a aggregate efficiency value control to use (e.g., the minimum, maximum, average, sum, etc.), including an option to ignore seized resource efficiencies.

If using the 'Switch Resources If Possible' *Off Shift Rule* at a Server, Combiner, or Separator object, Simio automatically adjusts the remaining processing or setup time to reflect efficiency changes due to switched secondary resources.

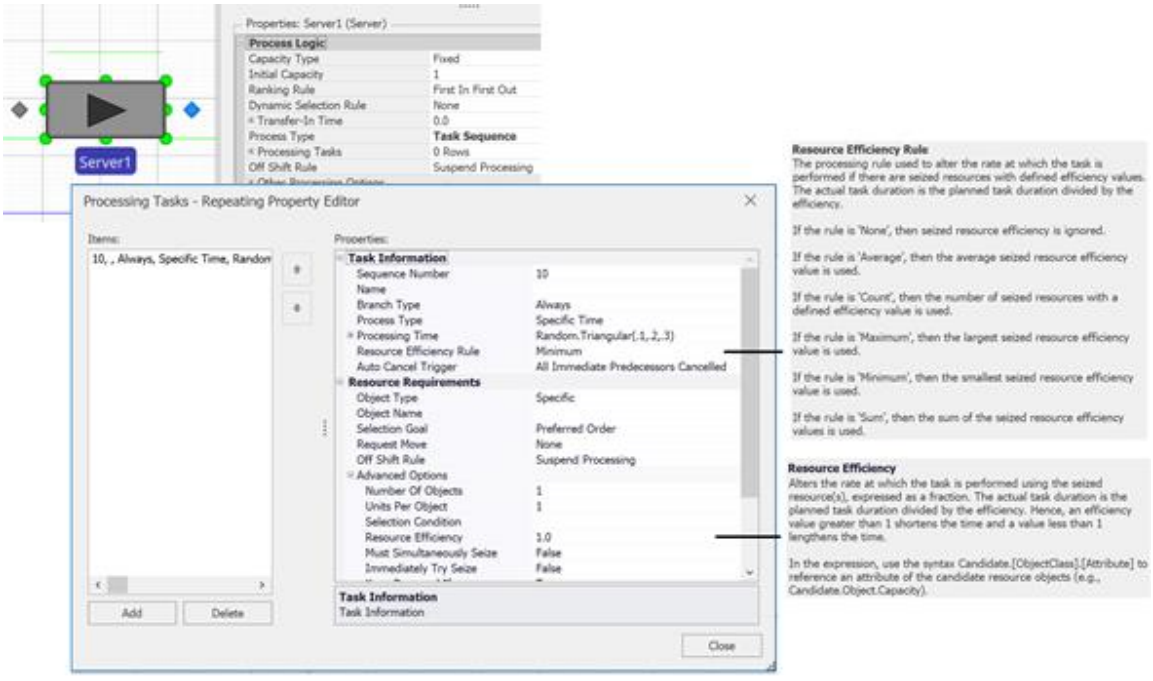
Resource Efficiencies Details– Server/Combiner/Separator

Within the Secondary Resource for Processing as well as the Process Tasks areas, the *Resource Efficiency Rule* and *Resource Efficiency* properties have been added.

If the *Process Type* is 'Specific Time', the *Resource Efficiency Rule* is visible within the Other Processing Options of Process Logic area. This property is used to alter the rate at which work is performed if there are seized resources with defined efficiency values. The actual work duration is the planned work duration divided by the efficiency. The *Resource Efficiency* property is used within the Secondary Resources (Resource for Processing) area to alter the rate at which the task is performed using the seized resource(s), expressed as a fraction. An efficiency value greater than 1 shortens the time and a value less than 1 lengthens the time.



If the *Process Type* is 'Task Sequence', the *Resource Efficiency Rule* is visible within the Task Information area of the repeatable processing tasks. The *Resource Efficiency* property is specified within the Resource Requirements area for the task. Note that *Resource Efficiency* can be specified for each resource required if multiple resources have been referenced within a data table.



<u>Resource Efficiencies Details– Delay and Changeover Steps</u>

The Delay and Changeover steps have both been enhanced to allow for dynamic adjustments to the delay or setup time. A *Delay Multiplier* expression type property has been added to each of these steps to increase or decrease the delay time or total setup time due to the changeover by some factor. Note that expression is automatically reevaluated if the process is suspended and resumed, potentially adjusting the remaining delay or setup time.

<u>Resource Efficiencies Details– Seize Step</u>

The Seize step has been enhanced to include an optional *Resource Efficiency* value that can alter the rate at which work is performed using the seized resource(s), expressed as a fraction. The actual work duration is the planned work duration divided by the efficiency.

Resource Efficiencies Details– New Functions

We have added three (3) new functions to allow the user to access the aggregate efficiency of resource(s) that have been seized.

Object.SeizedResources.AggregateEfficiency(type) - Calculates and returns an aggregate efficiency value for the list of resources currently seized by the object.

Task.SeizedResources.AggregateEfficiency(type) - Calculates and returns an aggregate efficiency value for the list of resources currently seized by the task's associated object, filtered to only include the resource seizes that apply to the task's execution.

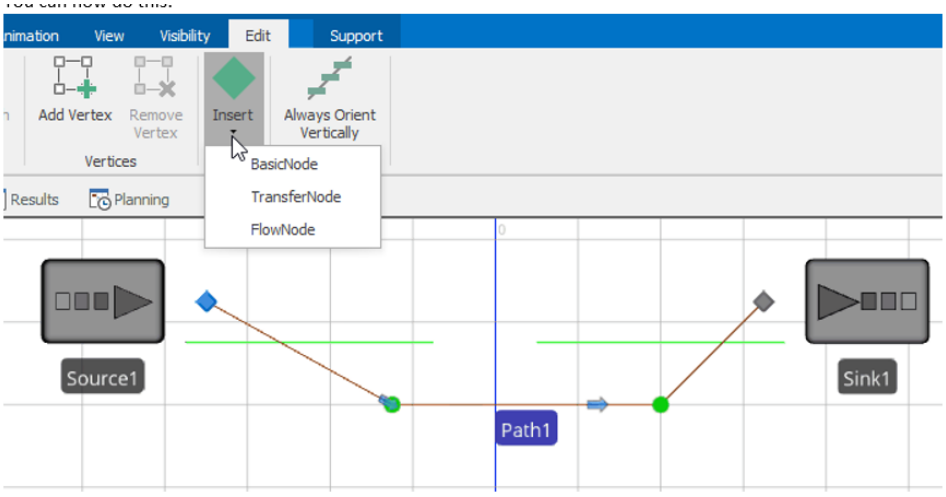
Token.SeizedResources.AggregateEfficiency(type) - Calculates and returns an aggregate efficiency value for the list of resources currently seized by the token's associated object, filtered to only include the resource seizes that occurred specifically due to the token's execution.

For all of the above functions, the aggregate type is an integer argument with possible values: 0 = None, 1 = Average, 2 = Count, 3 = Maximum, 4 = Minimum, 5 = Sum. Note that if the aggregate type is None or if there are no seized resources with defined efficiency values, then the value NaN is returned.

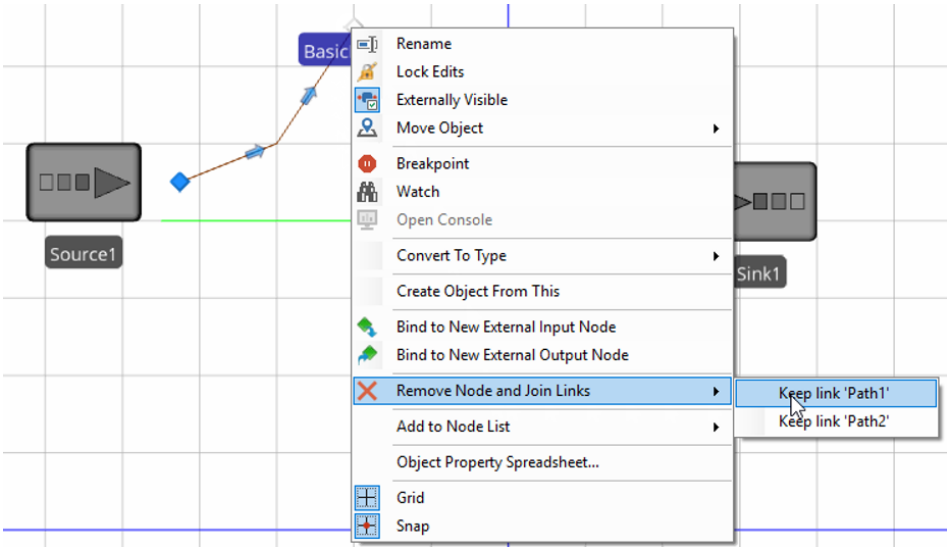
Adding and Deleting Nodes from Links

We have added the much-requested feature of being able to add or delete one or more nodes from an existing link.

As you can see from the below diagram, there is an existing Path link between Output@Source1 and Input@Sink1. By clicking on the path, the user is directed to the Edit menu which will allow for the addition (Insert) of any type of node to the path.



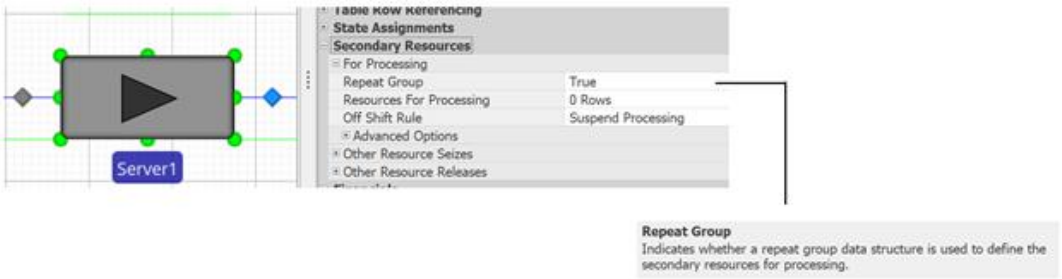
Alternatively, if the path has one or more existing nodes that are no longer required, the user may right click on the node (as in BasicNode1 below) and select 'Remove Node and Join Links', then select the path that will remain as the current link (e.g, Path1 below).

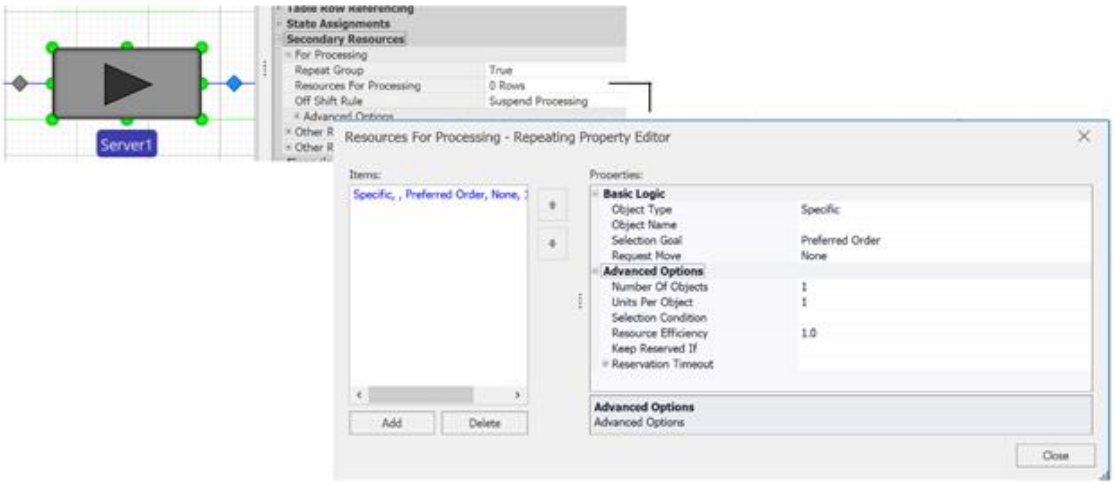


Secondary Resources Enhancements

We have enhanced the secondary resources functionality within the Server, Combiner, and Separator in the Standard Library and Filler and Emptier in Flow Library. These enhancements allow for more flexibility in specifying either single or multiple secondary resources for processing or before/after processing.

These enhancements include having multiple *Secondary Resources for Processing* (resources used throughout the entire processing of an entity) to be specified using a repeat group data structure. If using a repeat group to specify *Secondary Resources for Processing*, we also provide *Off Shift Rule*, *Must Simultaneously Seize*, *Immediately Try Seize*, and *Immediately Try Allocate When Released* property values to be specified. If not using a repeat group to specify *Secondary Resources for Processing*, allow *Number Of Objects*, *Units Per Object*, and *Must Simultaneously Seize* property values to be specified for the single resource requirement.





For each *Secondary Resources -> Other Resource Seizes* repeat group, we now include a *Resource Efficiency* property for a resource seize requirement (inside repeatable property). See the above release notes on new efficiency features.

For each *Secondary Resources -> Other Resource Seizes* repeat group, we have added an *Immediately Try Seize* property value to be specified. Similarly, for each *Secondary Resources -> Other Resource Releases* repeat group, we now include an *Immediately Try Allocate When Released* property value to be specified. These features were added to Simio for the main resource and task resources several sprints ago.

If using a 'Smallest Value' or 'Largest Value' *Selection Goal* is used to seize a secondary resource, we now display a *Value Expression* (like Selection Expression) in the properties window.

New Process Element Function

We have added a new function for the process element. This function was added to support implementation of the *OnSecondaryResourceCapacityChanged* process logic in the Standard Library objects when multiple secondary resources for processing have been seized by a particular entity, and the 'Switch Resources If Possible' off shift rule is being used.

TokensInProcess.NumberLinkedToObject(object) - Returns the number of tokens in process whqose associated (primary) object reference is a specified object.

New Scheduling Examples

We have added two new examples to the Simio installation. Please note these examples are meant to show specific concepts and include basic object animations. We will follow-up within the next sprint or two with more detailed documentation. They include:

SchedulingLaborEfficiencies – This example is like the SchedulingDiscretePartProduction example, in that it uses various tables to generate the objects within the Facility window. There are 5 employees (worker objects) that use different work schedules, as defined in table based repeating work schedules. The new resource efficiency properties are specified within Labor table, based on which resource is being used. The task sequences within the Servers indicate the worker list, resource efficiency and

selection rule based on largest efficiency. The model also uses the 'Switch Resources If Possible' *OffShift Rule* for workers.

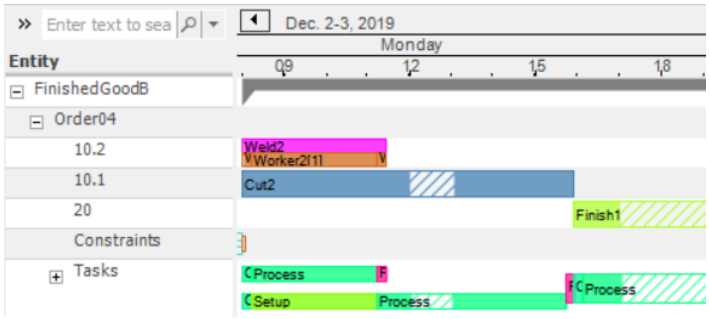
SchedulingParallelRouteController – This example is a modification of the SchedulingDiscretePartProduction example. It is intended to show an option for easily modeling parallel routing steps.

The Routings table has been modified to include **parallel production steps** within the routings of FinishedGoodA and FinishedGoodB. A 'route controller' Server object and it's referenced process titled 'PopulateImmediateSuccessorSequences' are used to generate entities for the various parallel process steps.

For example, in the below routing, the Sequence Number is used in the process logic to determine the parallel activities within the routing – Cut and Weld (10.1 and 10.2) operations are done in parallel before Finish (20) and then Ship (40). The Entity Workflow Gantt shown below shows the resources and sequence for the order, as they are processed.

Resources | Routing Destinations | Materials | Material Lots | Manufacturing Orders | **Routings** | Bill Of Materials | Work In Process | Manufacturing Orders Output

	Routing Key	Sequence	Material Name	Sequence Number	Route Number	SetupTime (Hours)	ProcessTime (Hours)
1	FinishedGoodB_10	Cut	FinishedGoodB	10.1	10	3	0.35
2	FinishedGoodB_20	Weld	FinishedGoodB	10.2	20	0	0.3
3	FinishedGoodB_30	Finish	FinishedGoodB	20	30	0	0.46
4	FinishedGoodB_40	Ship	FinishedGoodB	40	40	0	0.1



New SimBits

We have added three SimBits to our extensive list of small models that illustrate specific concepts. They include:

ServerBlockingApproaches – This project includes four models that demonstrate various ways of modeling blocking, that is, holding an entity at a Server or processing area until there is room at the next Server or processing area. The methods used include Connectors, Add-on Processes with Wait step and built in events and Entity Destination Types of selecting from a list / custom routing group element.

InventoryReplenish – This project includes three models that demonstrates some of the inventory *Replenish Policy* options in Simio. This includes the 'Min/Max', 'Order-Up-To' and 'ReorderPoint/ReorderQuantity' rules. Both the *Review Period* options, including 'Timer Review Period' and 'Continuous Review Period', are shown in each of the models.

ResourceSelectionConcepts – This project includes four models that demonstrate the use of Secondary Resources For Processing. The Models increase in complexity and demonstrate methods for selecting resources based on attributes like cost, efficiency and skill.

New Run.FreezePeriod Function – RPS Users

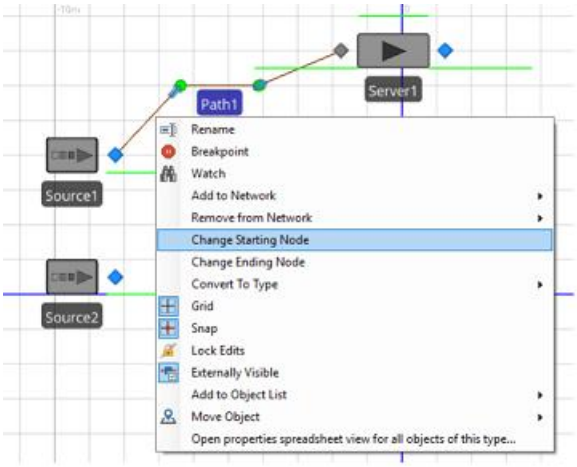
We have made the Run.FreezePeriod function now visible to RPS users. This function has been used in the SchedulingBatchBeverageProduction example. For more information, see the Freeze Period section of the documentation for the model and refer to the OnRunInitialized_FreezeSchedule process logic.

Simio Release 11 – Sprint 191 – June 27, 2019

Within this sprint, we have so many cool new features that have been requested by our users as well as our internal consultants. These range from being able to change the starting or ending node of a link (without losing the link!) and auto-creation of links to enhanced table functions and referencing for more flexibility to repeating table work schedules.

Link Enhancement - Change the starting/ending node of an existing link

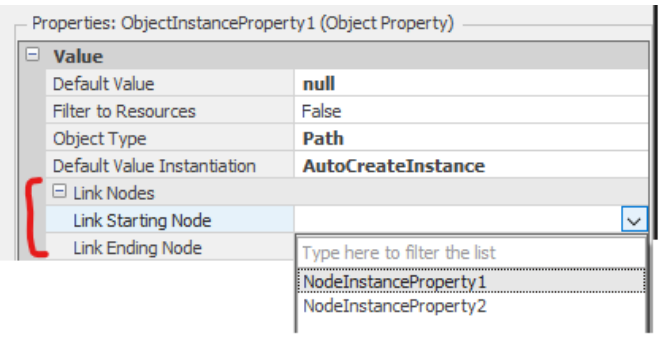
We now provide the much-requested feature of changing the starting or ending node of an existing link without losing the link or any vertices. Right clicking on an existing link will allow the user to Change Starting Node or Change Ending Node. When one of these is selected, the user has the ability to choose another node by clicking on it, at which point the link will disconnect from the original node and reconnect to the selected node. Multi-select of links also allows these two options.



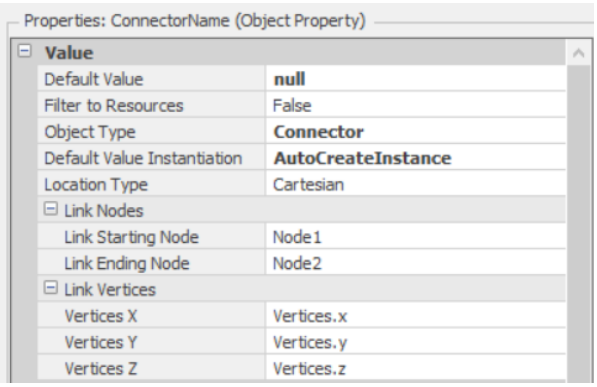
Links Enhancement - Auto Create

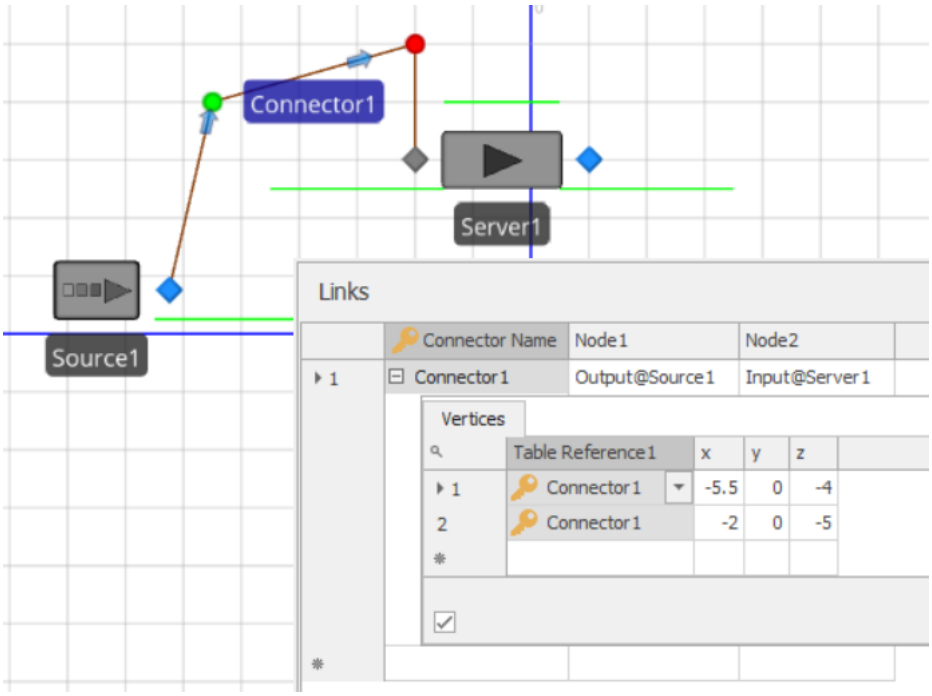
We have added the capability to auto-create links through data tables. Prior to this sprint, we've had the capability to auto-create objects (from a table) within the Facility window, such Server, Worker, Vehicle, TransferNode, etc. Now, we have a similar structure for generating links in the Facility window by specifying the starting and ending nodes.

If the *Default Value Instantiation* is 'AutoCreateInstance' and the *Object Type* is a link type (or an Object Type property), then the *Link Starting Node* and *Link Ending Node* appear in the property grid:



Users can also specify a detail table column (or static value) for the position of the vertices of an auto-created link in a table. The example below shows two tables, Links and Vertices. The Links table creates the connector links between the node names listed in Node1 and Node2 columns. The related Vertices table lists the x, y, and z cartesian coordinates of the vertices of the connector(s). *Location Type* can alternatively be 'Geographic'.





New Table Function - **TableName.KeyColumn.RowForKey(keyValue)**

We have added a new table function, **TableName.KeyColumn.RowForKey(keyValue)** that returns the row index of the row for which the specified KeyColumn has the specified keyValue (or zero if no row is found). It's important to note that this function respects master-detail relationships, and only looks at the detail row subset for a given master row selection (and returns the row index as related to that subset, not the overall index).

Thus, the user could have a table, Resources, as below, with the ResourceName as the key column. The following expression, Resources. ResourceName.RowForKey(Weld1) would return the value '8'.

Resources | Routing Destinations | Materials | Material Lots | Manufacturing Orders | Routings | Bill Of Materials | Wo

	Resource Name	Description	XLocation (Meters)	ZLocation (Meters)	Object Type	Work Schedule
1	Cut	Cut	-8	5	SchedTransferNode	StandardWeek
2	Cut1	Cut1	-4	2	SchedServer	StandardWeek
3	Cut2	Cut2	-4	7	SchedServer	StandardWeek
4	Shape	Shape	5	-1	SchedTransferNode	StandardWeek
5	Shape1	Shape1	1	-2	SchedServer	StandardWeek
6	Shape2	Shape2	9	-2	SchedServer	StandardWeek
7	Weld	Weld	5	11	SchedTransferNode	StandardWeek
8	Weld1	Weld1	1	12	SchedServer	StandardWeek
9	Weld2	Weld2	9	12	SchedServer	StandardWeek
10	Finish	Finish	12	5	SchedTransferNode	StandardWeek
11	Finish1	Finish1	15	2	SchedServer	StandardWeek
12	Finish2	Finish2	15	7	SchedServer	StandardWeek
13	MoArrivals	MoArrivals	-14	5	SchedSource	StandardWeek
14	Resource1	Resource1	-14	-3	SchedResource	StandardWeek
15	Resource2	Resource2	-14	1	SchedResource	StandardWeek
16	Ship	Ship	20	5	SchedSink	StandardWeek

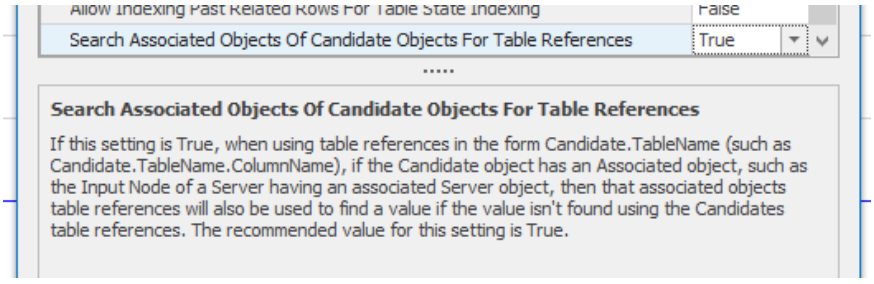
The function could alternatively be used to access other information within the Resources table for a given key value. Resources[Resources. ResourceName.RowForKey(Weld1)].ObjectType would return 'SchedServer'.

The keyValue may also be an expression that holds the information, such as an entity object state, 'ModelEntity.WhichWorkstation'.

Previously, the Search step could be used to access the row information. This new function method is faster than using Search step to access the same information, especially with large data table searches.

Table Referencing Enhancement – Using Associated Objects for Candidate Objects Table References

There is a new compatibility bit "Search Associated Objects Of Candidate Objects For Table References"



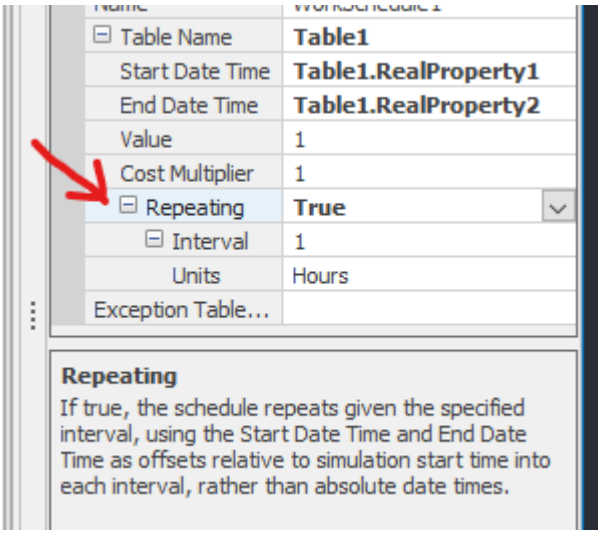
Default value of this setting is 'False' for models built prior to Sprint 11.191, 'True' for new models built in Sprint 11.191+.

In the past, when using the Candidate.TableName.ColumnName, for example, within an *Entity Destination Type > Selection Condition*, the candidate column required node name references. Many times, however, the data within a table may include the associated object names (Server names) and not the node names. Thus, as an enhancement when using the Candidate.TableName.ColumnName nomenclature, Simio will now search for either the object or associated object to find a value. This includes Search steps and any other construct where Candidate.TableName.ColumnName can be specified.

As an example, the user could have a TransferNode that selects the *Entity Destination Type* from a list of nodes. The options in the list contain multiple nodes that are associated with Servers. There is a table, named Resources, that includes the Server names and some associated columns such as Cost, WorkSchedule, etc. The Selection Condition on the TransferNode can now reference 'Candidate.Resources.Cost <= 30' where the node names are not listed in the Candidate table, but the associated Servers and their cost column are evaluated as the candidates for selection.

Repeating Table Based Work Schedules

There is a new *Repeating* property for Table Based Work Schedules that, if 'True', will repeat every specified time interval. For example, a user may define a 1 week work schedule within a table that will repeat every 7 days. Exceptions, such as holidays, can be specified within a referenced Exception Table.

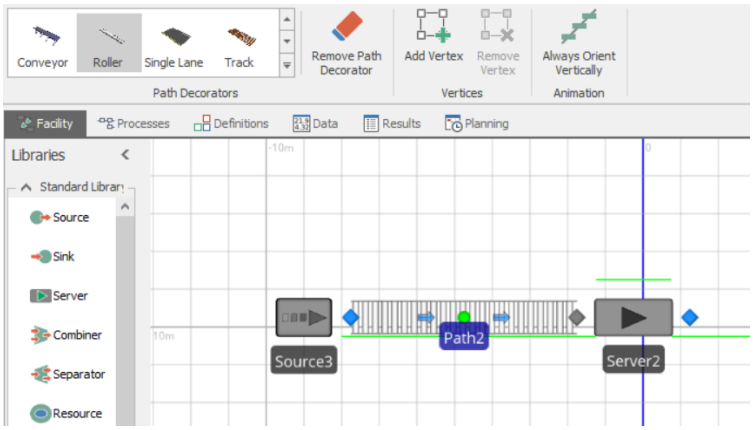


It's important to note that if the table schedule Start Date is AFTER the simulation start date, Simio will assume 0 capacity (no schedule specified in table) for the first X amount of time from simulation start date through schedule start date, then repeating after Interval time.

As an example, let's say the simulation start date is 6/09/19 and the schedule starts 6/10/19 and contains a single day of schedule/capacity times. If the repeatable interval is 1 day, it will include 6/09/19 of 0 capacity and then repeat, thus the 6/10/19 schedule will not be realized. If the repeatable interval is 2 days, it will include 6/09/19 of 0 capacity, then 6/10/19 schedule from table and then repeat.

New Path Decorator – Roller

We have added a new Roller type path decorator that can be used with any of the links in Simio, including Conveyors and Paths.

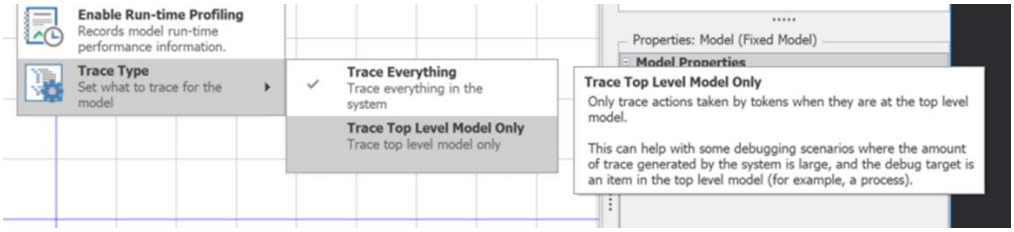


Simio Release 11 – Sprint 190 – June 4, 2019

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.

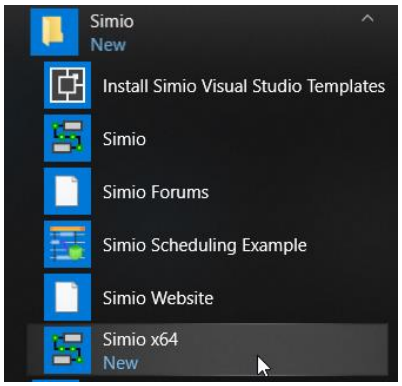
Trace Enhancement

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

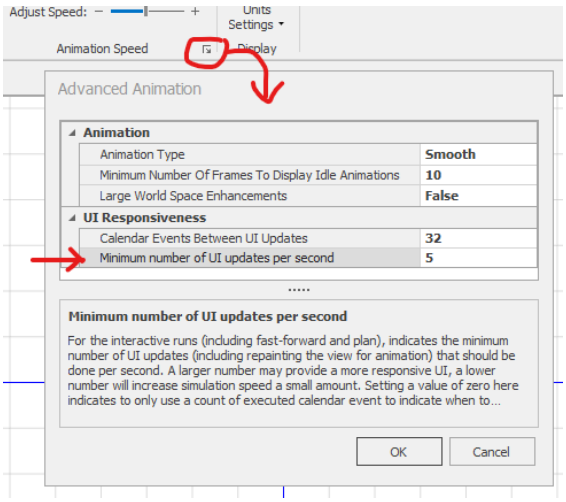
We have updated our Sketchup import to support Sketchup 2019.

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

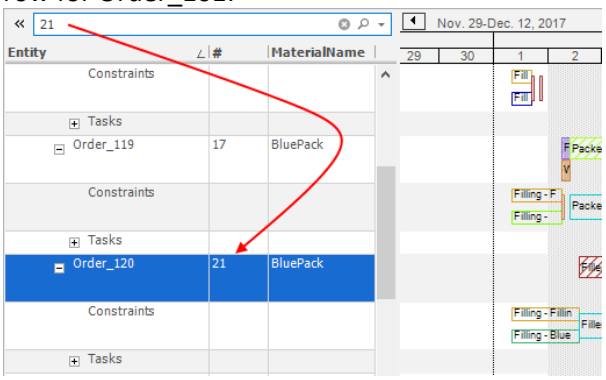
There is a new “Minimum number of UI updates per second” entry in the Advanced Animation dialog. It’s defaulted to 5 and provides much better UI responsiveness for otherwise unresponsive models.

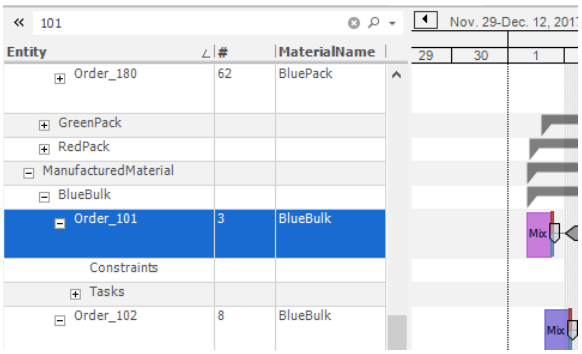


Gantt Views – Search and Target Details

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:

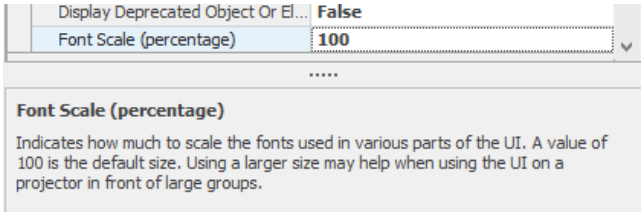




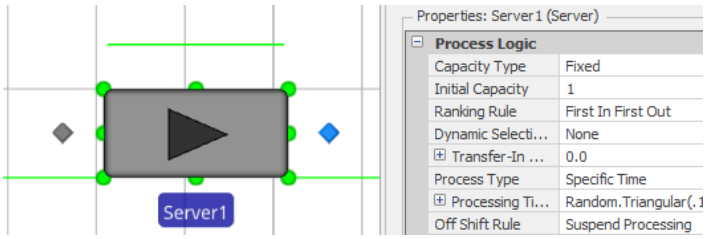
Also within the Planning tab, if a user double-clicks on an order/entity within the Target Detail tab, the user is taken to the same order within the Entity Workflow Gantt.

Font Scale for Presenting/Teaching/Projection Scenarios

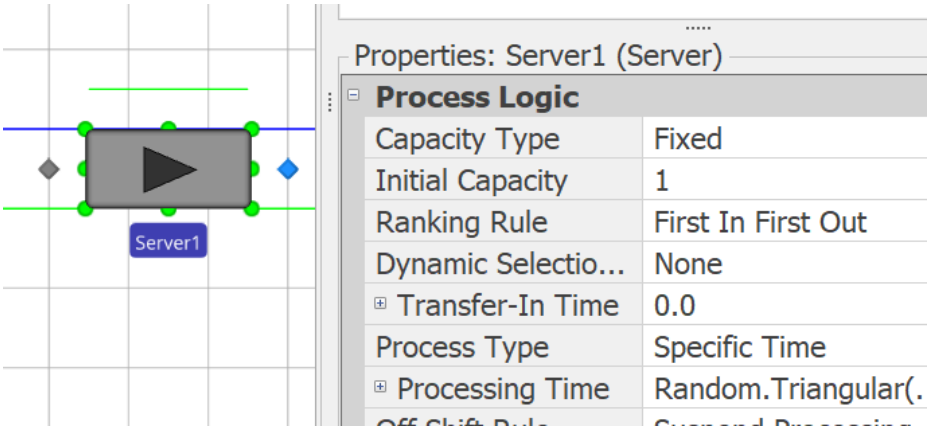
There is a new Font Scale setting in File -> Settings allowing you to scale up the font used across the application:



So normally the UI looks like:



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



Modeling Helper Add-Ins – RPS Only

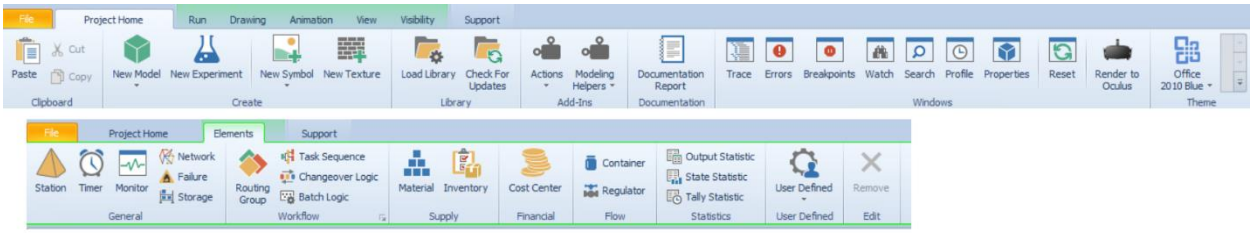
We have added the following new events that support modeling helpers when running Simio as an 'embedded' solution via the SimioDLL.dll library:

Tables Importing
Table Importing
Table Imported
Tables Imported
Plan Run Starting
Plan Run Ended

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!



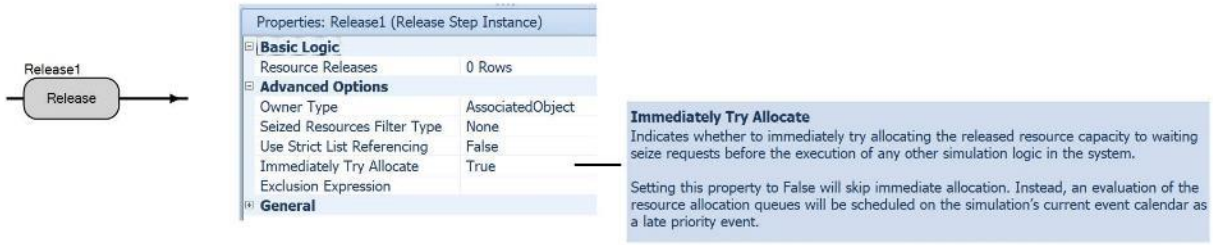
New SimBit
DashboardsWithinExperiments – This example is a modification to the SimBit ServerUsingTaskSequenceWithWorkers and includes an experiment where the number of workers and other properties can be changed. The new Dashboard Reports feature within Experiments has been used to include graphical 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).

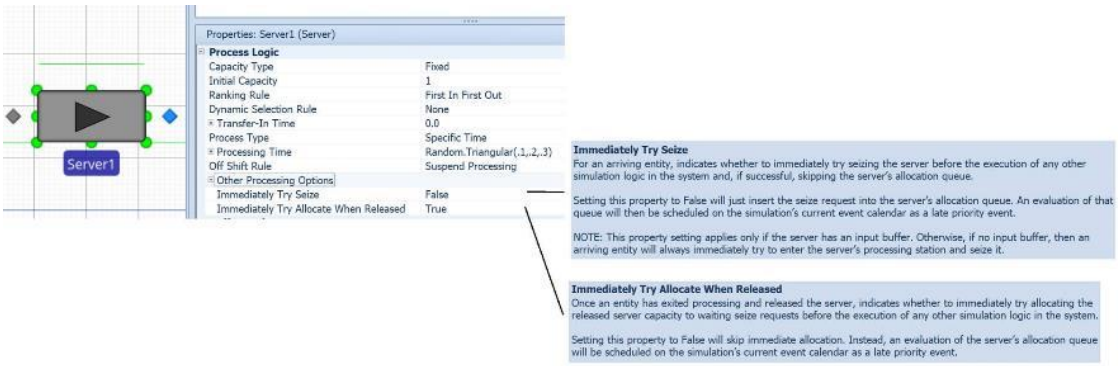
This feature is useful when multiple resources may be released simultaneously, as setting this property to False will allow entities in resource allocation queues to wait until all resources have been released before attempting to allocate a resource to an entity(s) based on any resource selection rules.



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.

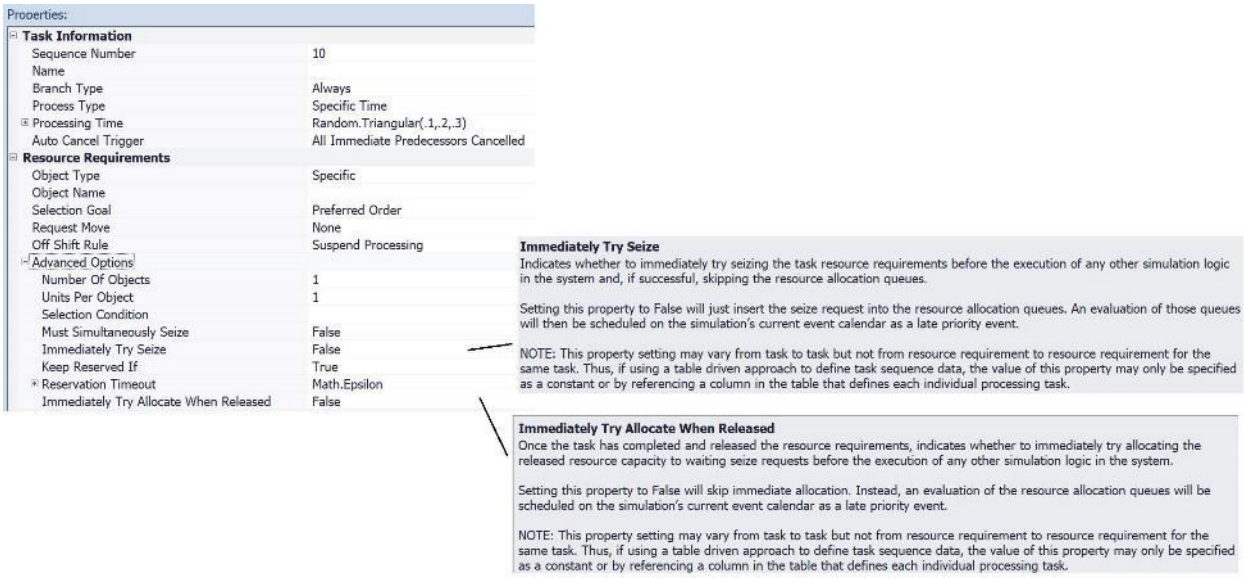


For all the same objects, there are also the same new properties within the *Secondary Resources -> Resource for Processing -> Advanced Options* category as they relate to the secondary resource.

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.

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.

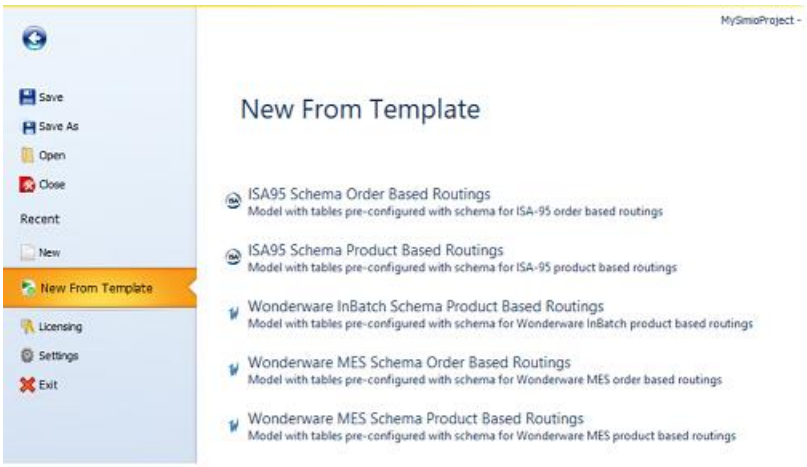

Properties:
Task Information
Sequence Number 10
Name
Branch Type Always
Process Type Specific Time
Processing Time Random.Triangular(.1,.2,.3)
Auto Cancel Trigger All Immediate Predecessors Cancelled
Resource Requirements
Object Type Specific
Object Name
Selection Goal Preferred Order
Request Move None
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.

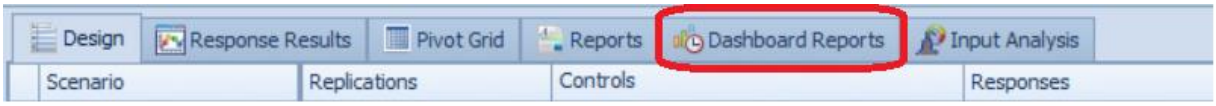
Trace Enhancement – Include RunTime Error Information

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.

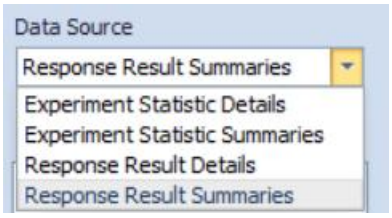
Input@Server1	OnEnteredToAssociatedObj...	1	[End]	Process 'Input@Server1.OnEnteredToAssociatedObject' execution ended.
Server1	OnEnteredProcessing	2	[Begin]	Process 'Server1.OnEnteredProcessing' execution started.
			[EndTransfer] Int...	Entity 'DefaultEntity.9' ending transfer into station 'Server1.Processing'.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



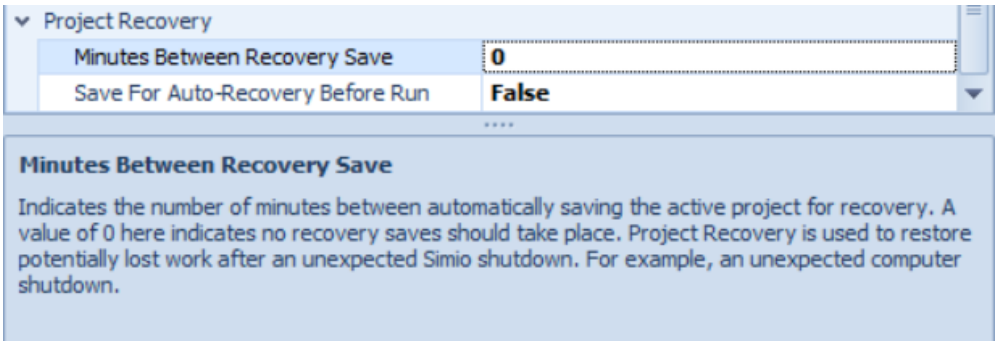
This tab opens a view like the Dashboard Reports view for interactive runs. Under Experiments, however, there are a different set of data sources, as shown below, where the “Summaries” data sources include average/half-with/min/max across replications, and the “Details” data sources have the individual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



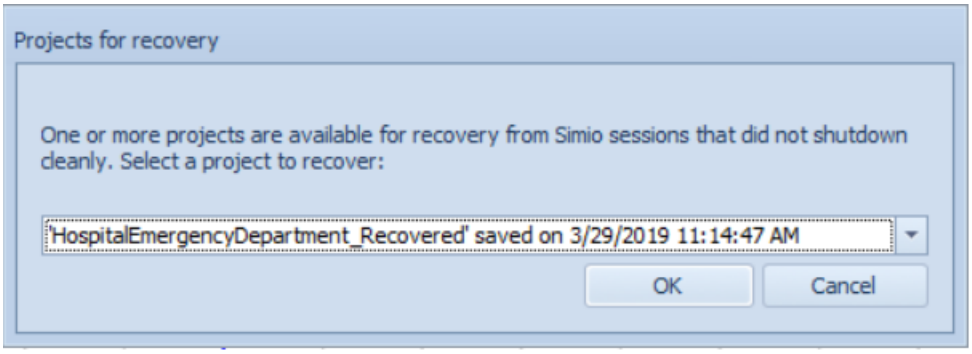
In the below graph, the HospitalEmergencyDepartment example was used, with additional Responses to capture ScheduledUtilization statistics for the Nurse and Doctor workers. Once the experiment scenarios were run, the Response Results Summaries were used to generate a ‘Chart’ type dashboard. Within the 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”:



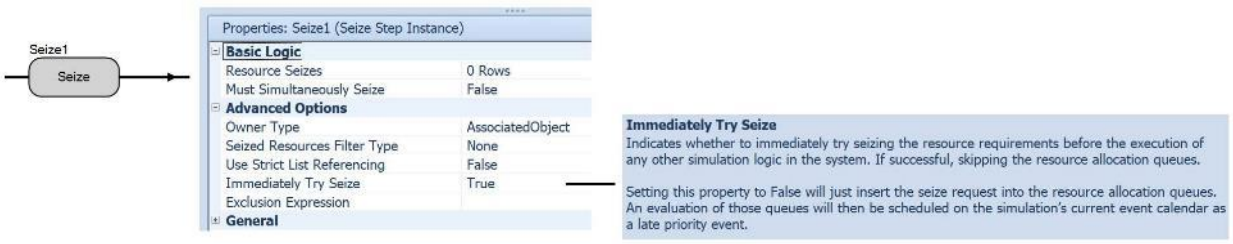
By default, this feature is off, as the *Minutes Between Recovery Save* is set to '0' and the *Save for Auto-Recovery Before Run* is set to 'False'. If a user was to turn one of those on to get a "Recovery Save" (wait the time limit or run), then end Simio using 'End Task' through the Task Manager (or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

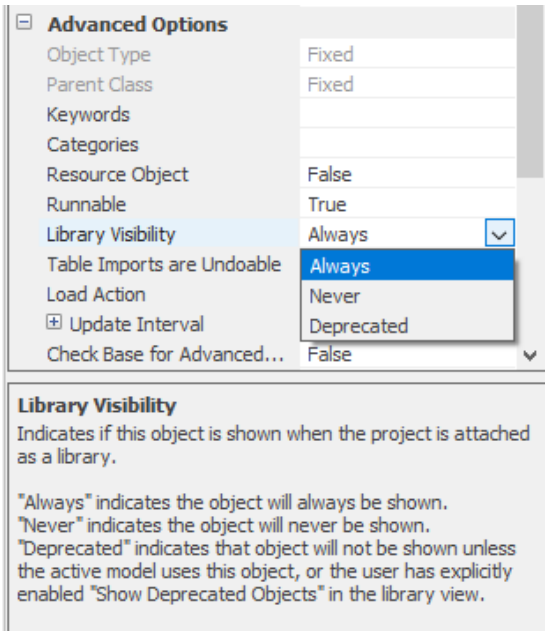
RelationalTablesInRepeatingProperty – This updated SimBit now uses Servers in place of Workstations for processing multiple patient entity types through a system. Relational data tables are used to specify different processing times, as well as multiple doctors and/or nurses for processing, for the patient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

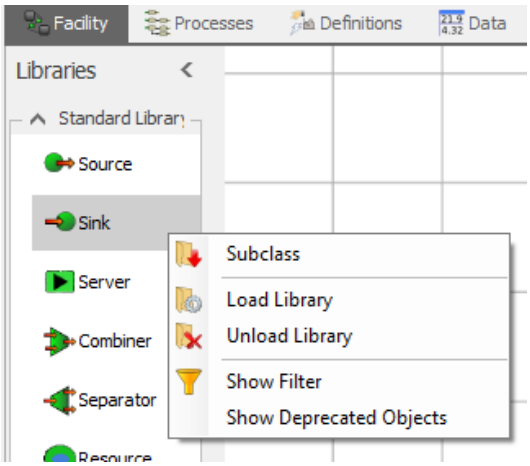
Workstation Object within Standard Object Library – Deprecated

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
Never
Deprecated
Load Action
Update Interval
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.

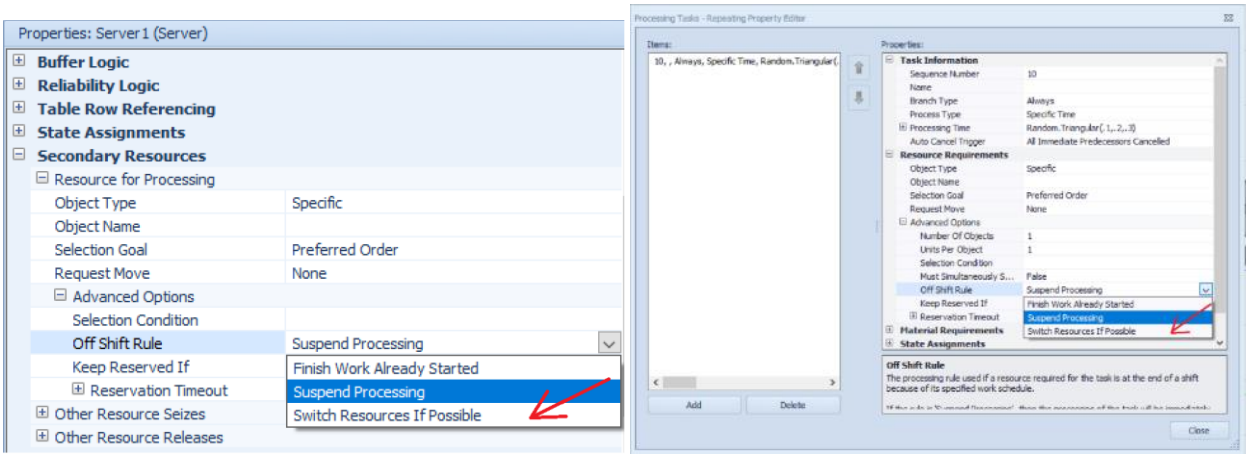

Facility
Processes
Definitions
Data
Libraries
Standard Library
Source
Sink
Server
Combiner
Separator
Subclass
Load Library
Unload Library
Show Filter
Show Deprecated Objects

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.

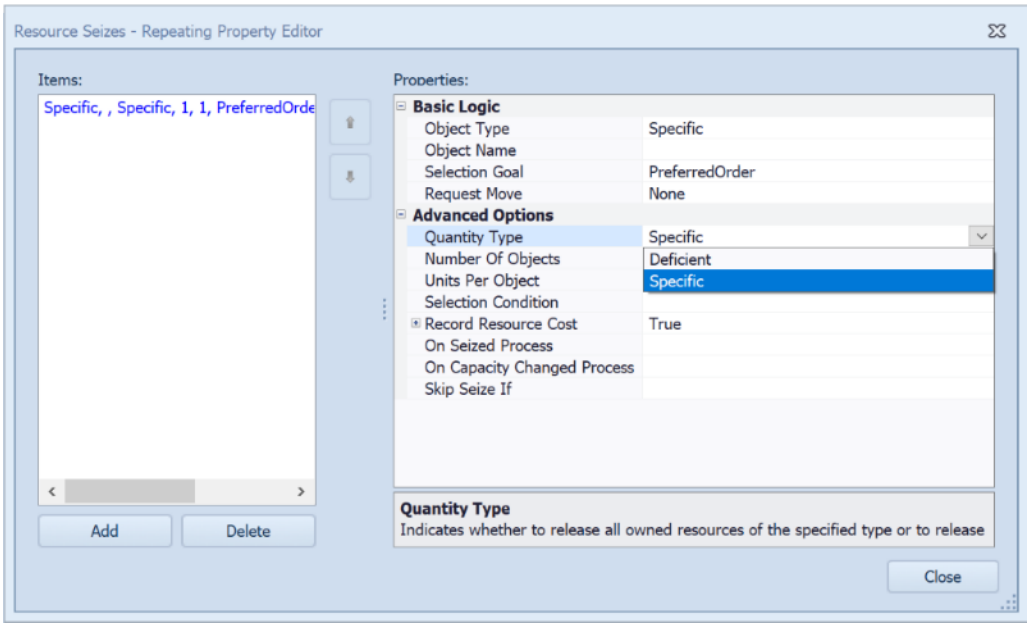
Task – New Function

We have added a new task function:

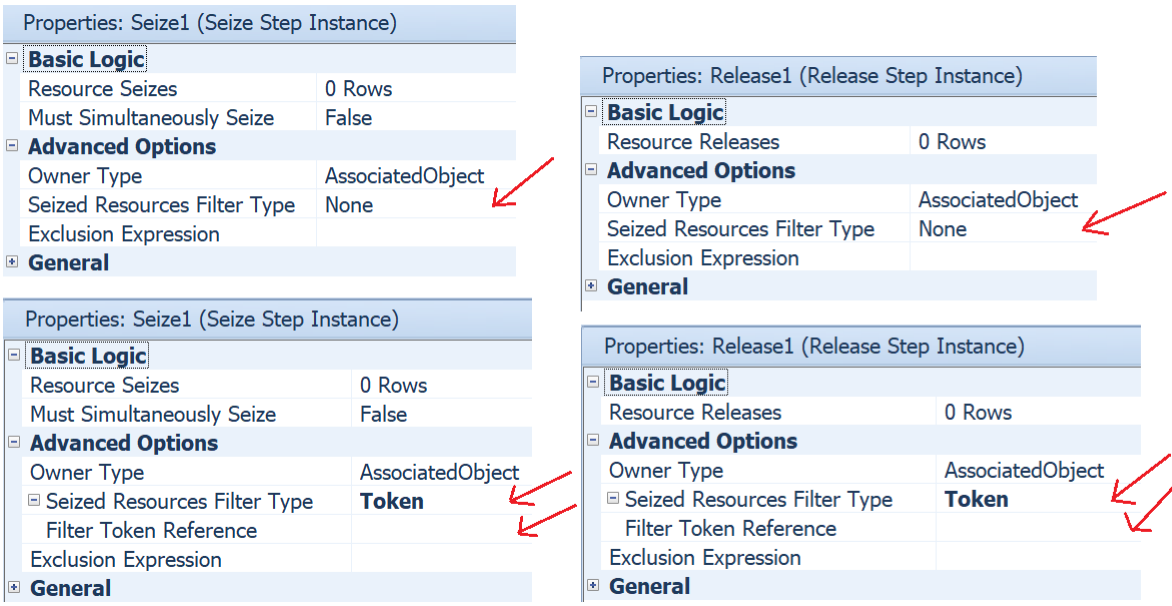
Task.PrimaryToken - Returns a reference to the token that is executing the task's primary process.

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'.

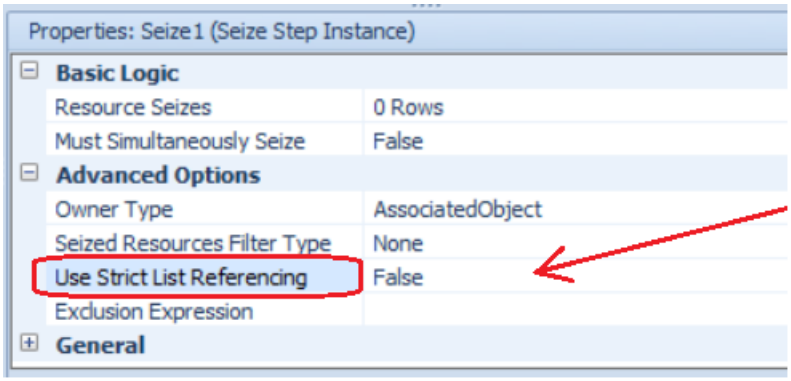


Then, within both the Seize and Release steps, we have added a *Seize Resources Filter Type* property that can be set to 'None' or 'Token'. With the 'Token' option, an additional property named *Filter Token Reference* can be specified. To see how these new properties are used, please review the 'OnSecondaryResourceCapacityChanged'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



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

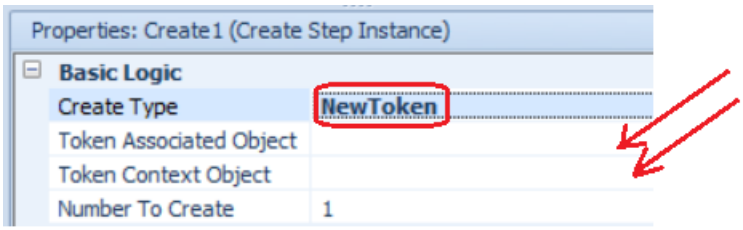
whether the released resource(s) must have been seized from the same specified list, instead of merely being members of the list. This is particularly useful when resources are members of multiple lists.



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

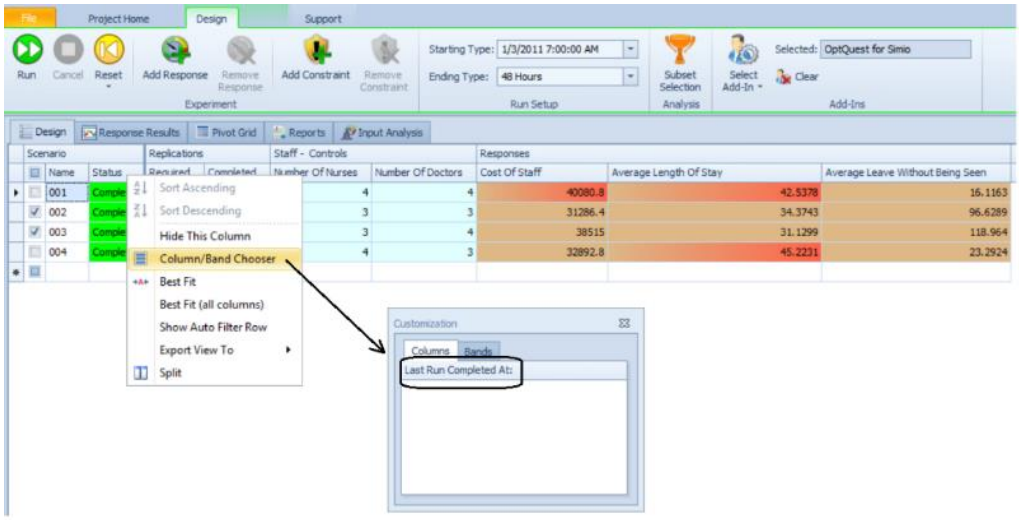
In Run Setup -> Advanced Compatibility Settings, an *Always Use Seize Selection Goal When Checking Resource Allocation Queue* compatibility setting has been added. This setting indicates whether a Seize step's selection goal should be used not only at the initial seize attempt but also whenever any resource 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.

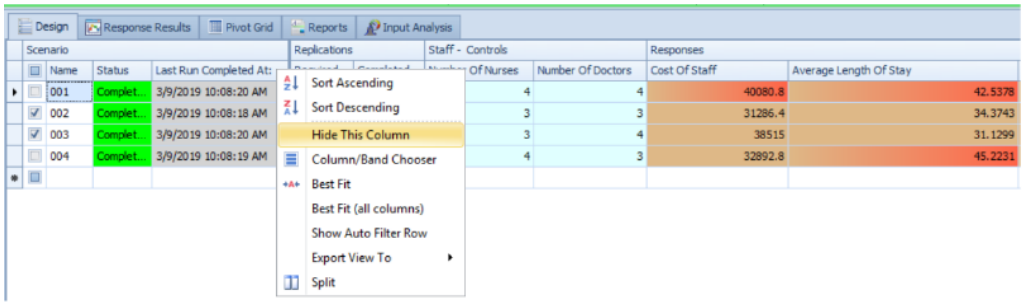
In the old software behavior – if the above compatibility setting is left at 'False' for an old model – whichever candidate resource's shift change happened to be processed first – usually based on the order that objects were placed in the model – was always the resource seized. The *Selection Goal* (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'.

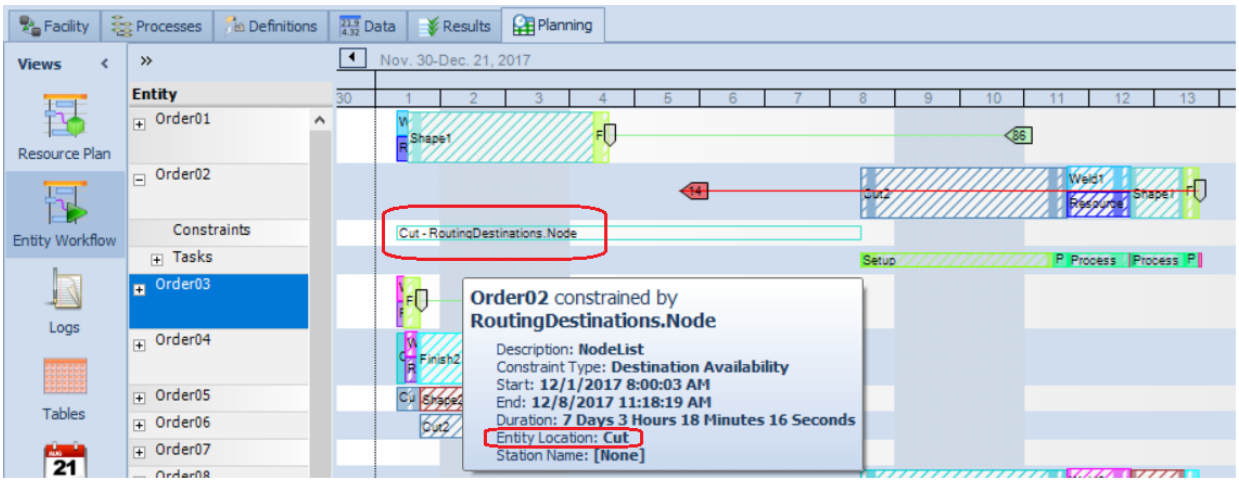


It now appears next to Status column. To hide the column, Right Click 'Last Run Completed At' and select 'Hide This Column'.



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.

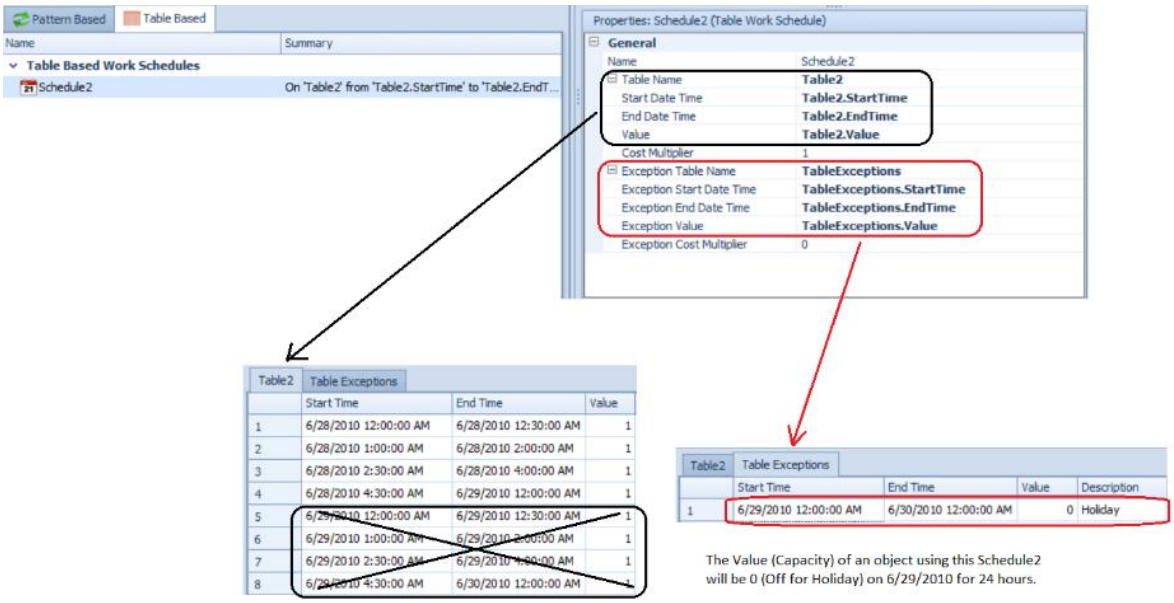
Simio Release 10 – Sprint 185 – February 18, 2019

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.

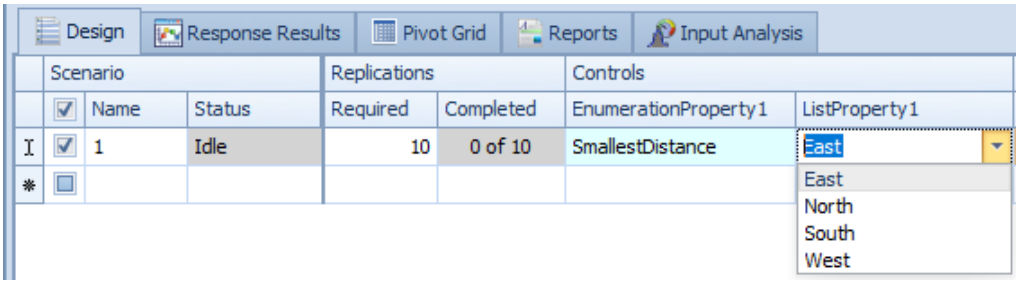
For example, in the below diagram, the data within the Exception table named TableExceptions (in red) will override the Table2 data for 6/29/2010 (in black with X).



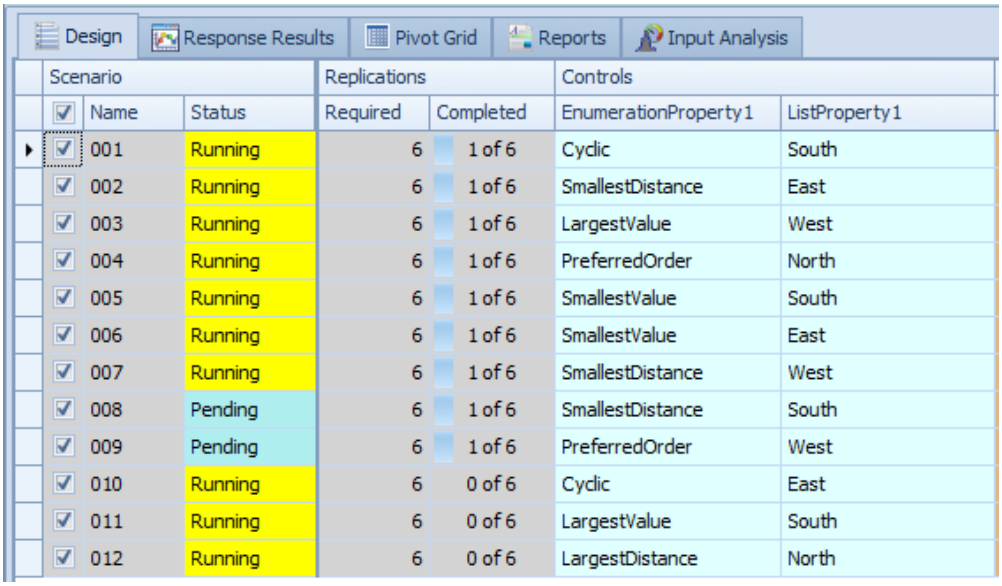
It's important to note that any Work Day Exceptions or Work Period Exceptions data specified within the object instance will take precedence over the Exceptions table data.

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.

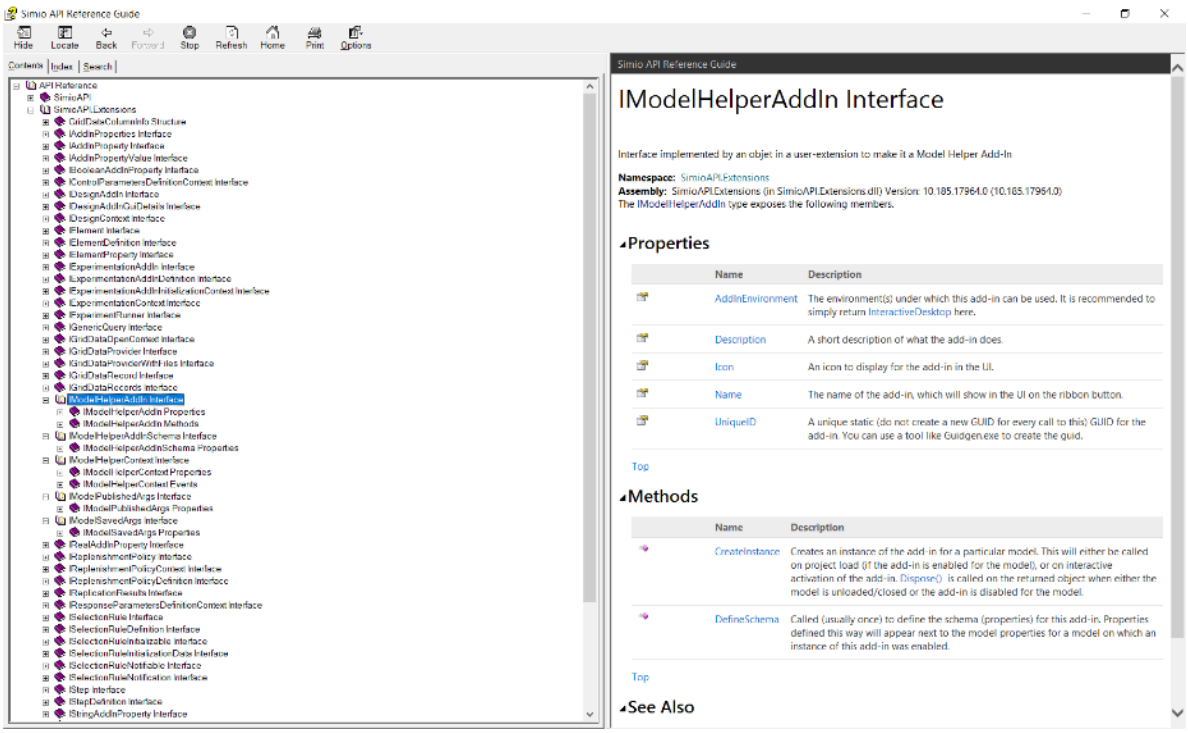


OptQuest will then simply choose different combinations of these values as part of the optimization.

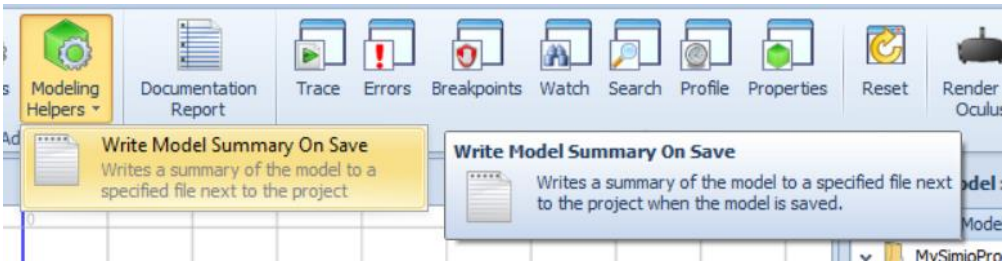


IModelHelperAddIn Interface

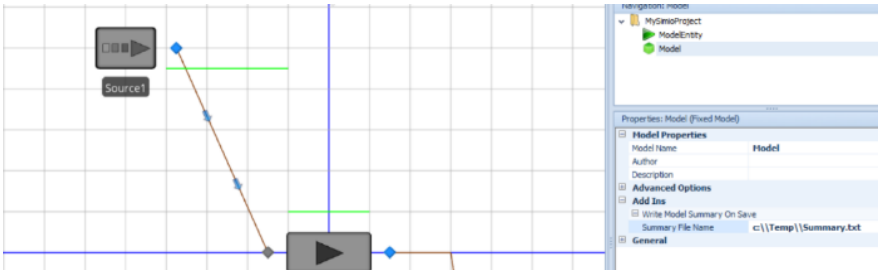
A new IModelHelperAddIn interface has been added to SimioAPI.Extensions. With this interface, there are methods to add code when the mode is loaded (CreateInstance) and when the model is closed (Dispose). There are also events handlers in the interface that can be used to add code when the model is saved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



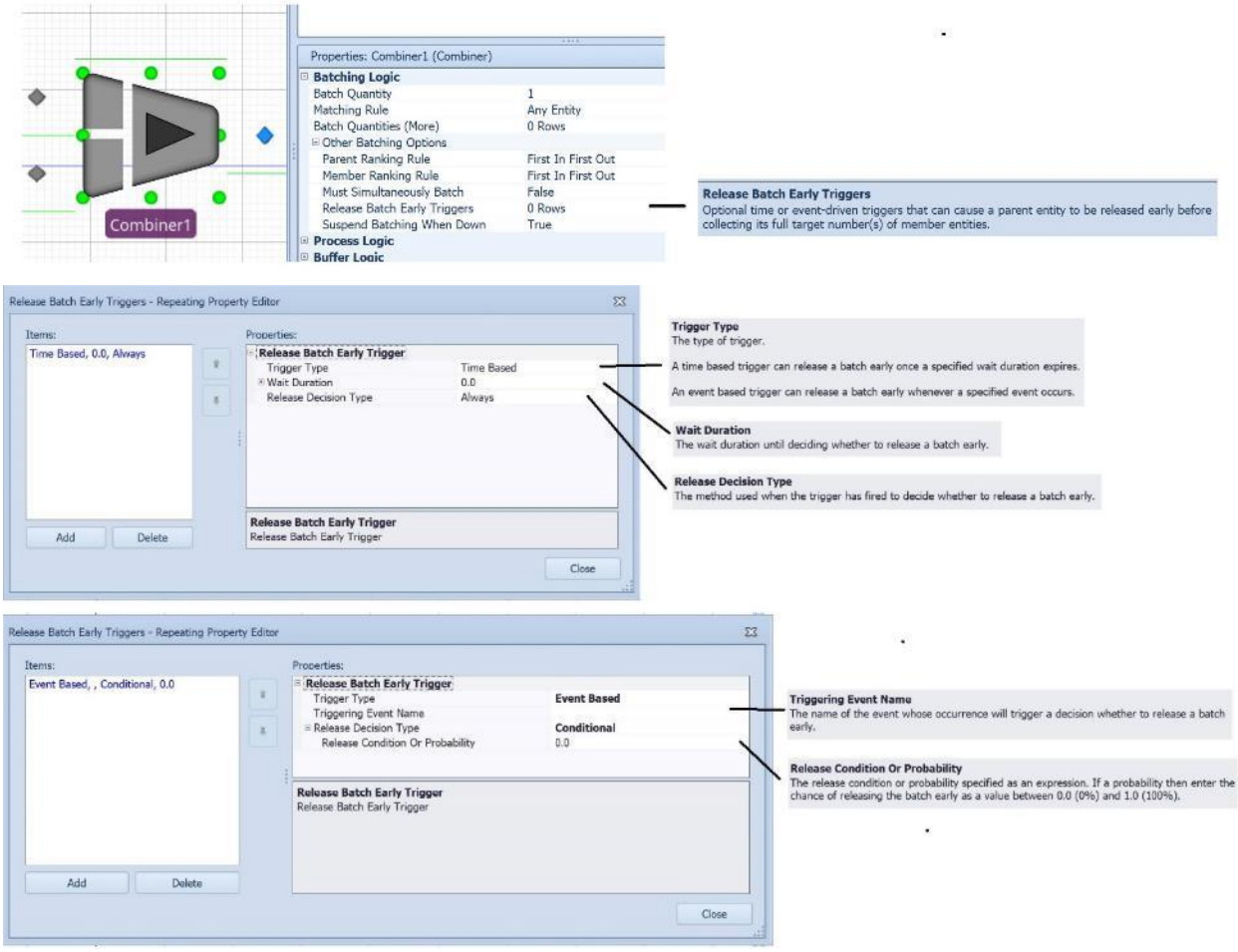
The file from this example is written after the model is saved.

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 *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.

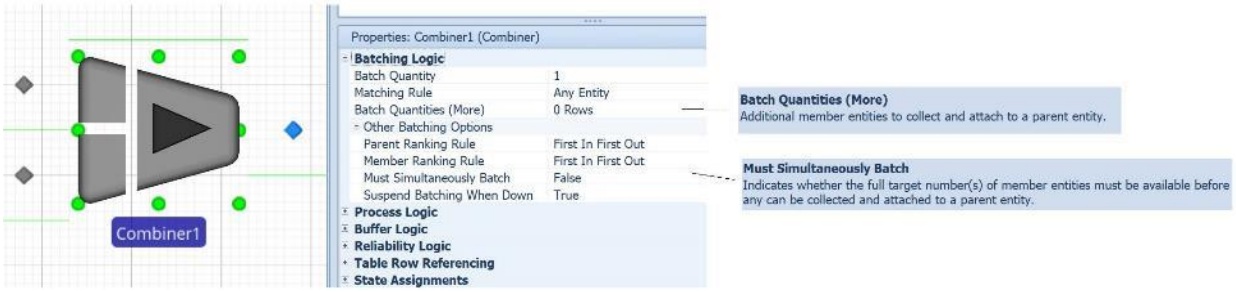
Inventory Element – Input Argument Support for Processes
Within the Inventory element, there are three different processes that may be specified, including *On Replenishment Order Process*, *On Balked At Backorder Process* and *On Reneged Backorder Process*. We have added support for Input Arguments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

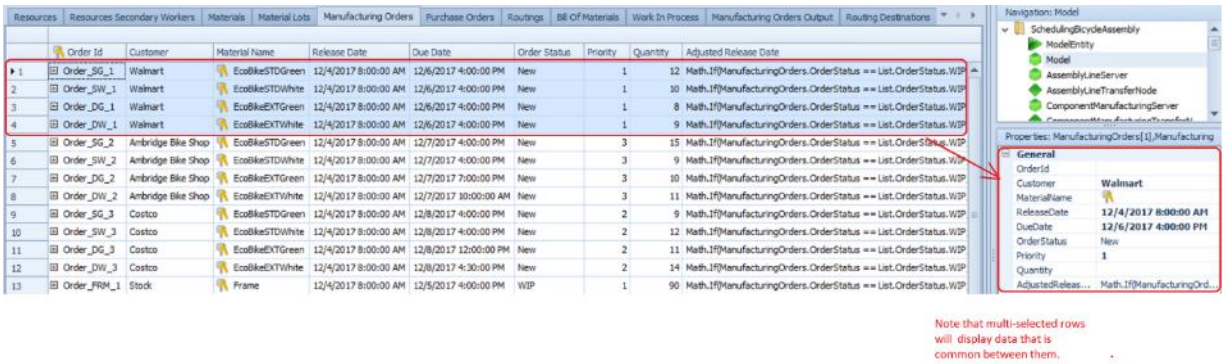
```
Entity 'A.29' waiting to be batched by batch logic 'Combiner1.Batching'.
Entity 'Pallet.28' waiting to collect '1' batch member(s) from queue state 'Combiner1.Batching.MemberQueue' using matching rule '[MatchMembers] ModelEntity.EntityType [AndParent] A [Matching Value = A]'.
Entity 'Pallet.28' waiting 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.

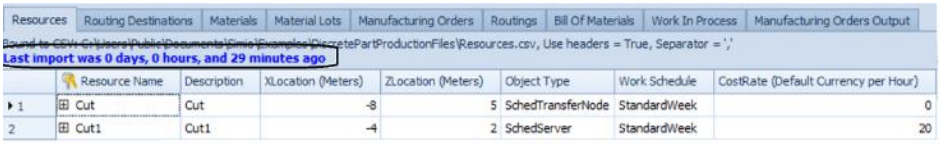
Data Table – Row Editing in Property Grid

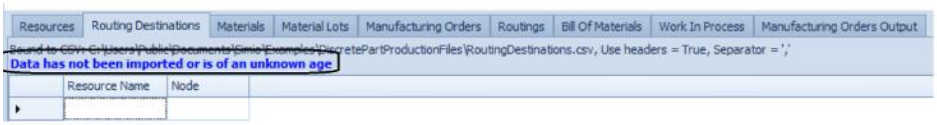
Within the Facility window, when a user selects an object or multiple objects, the property grid is provided for data input. We have now added this capability to **data table row editing**. When a single or multiple row are selected within a data table, the property grid now displays the column data 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.



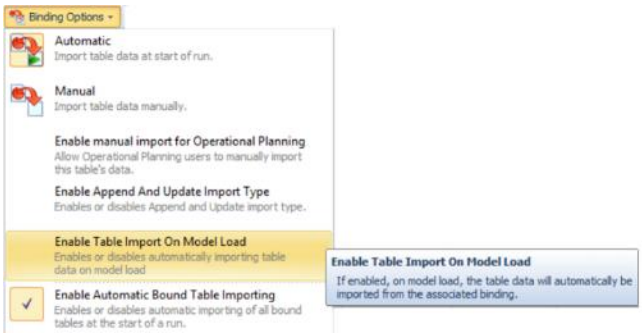
Data Table – Data Binding Import Enhancements

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.





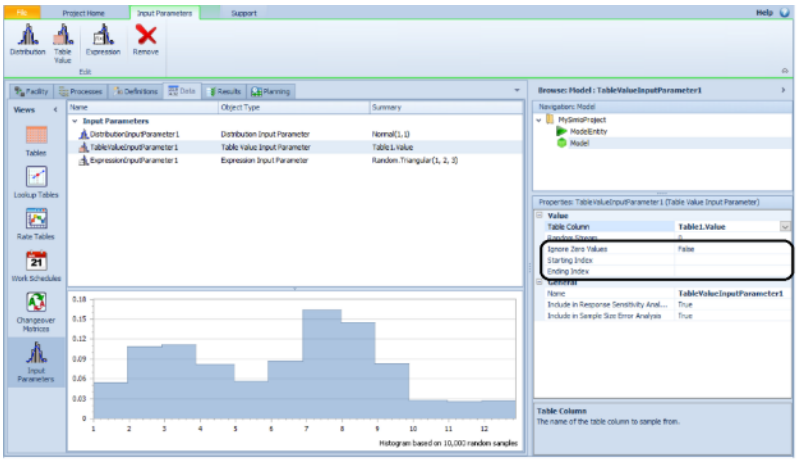
Also, within data tables, there is now an option to import the bound data upon loading the simulation model. This option is available with Simio RPS only.



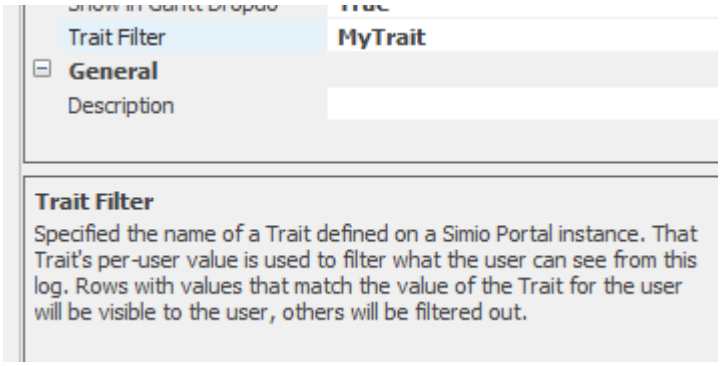
Input Parameters – Table Value Enhancements

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.



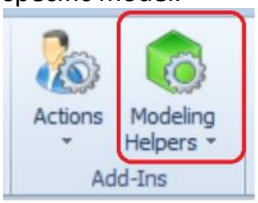
Resource Usage and Task Logs - Enhancement
There is a new *Trait Filter* property for the expression columns for the Resource Usage and Task logs:



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.

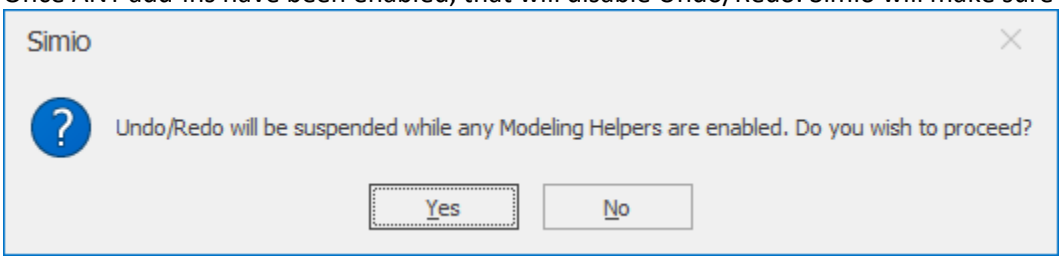
Project Home Ribbon – Modeling Helper Add-Ins – RPS Only
Users can now author design-time 'Modeling Helper' Add-Ins. These are pieces of code "bound" to a specific model.



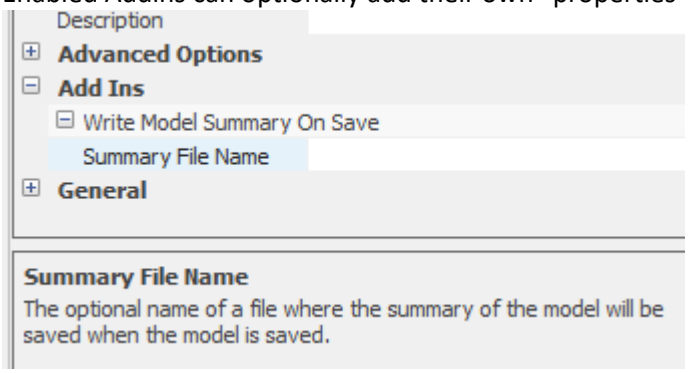
- The place a user starts to implement them is the SimioAPI.Extensions.IModelHelperAddIn interface.
- They get loaded with the model and unloaded when the model is closed.
- They have access to the full IModel design time API.
- They can subscribe to events on the model (right now only save).
- They 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

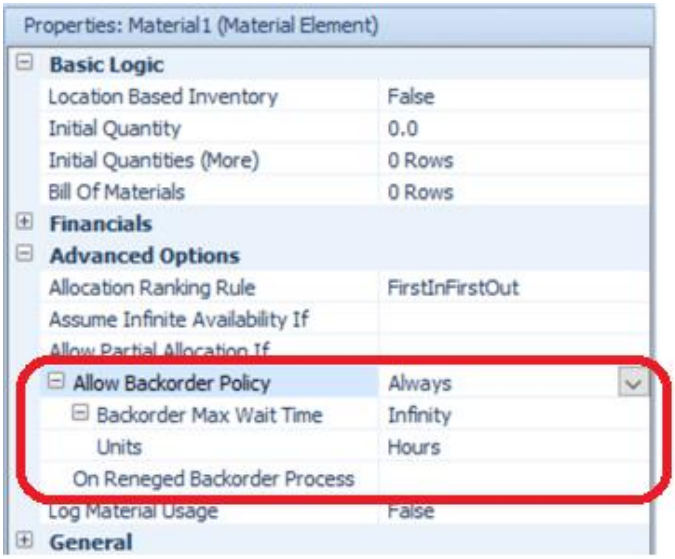
This sprint, we have completed our latest multi-sprint round of enhancements related to modeling supply chains -- specifically modeling materials and inventory. Our final set of enhancements provide enhanced inventory control and extensive statistics availability as well as extended support for backorders. 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 reneged backorders are described below. The Inventory and Material Backorders page within Simio help provides detailed information on this feature.



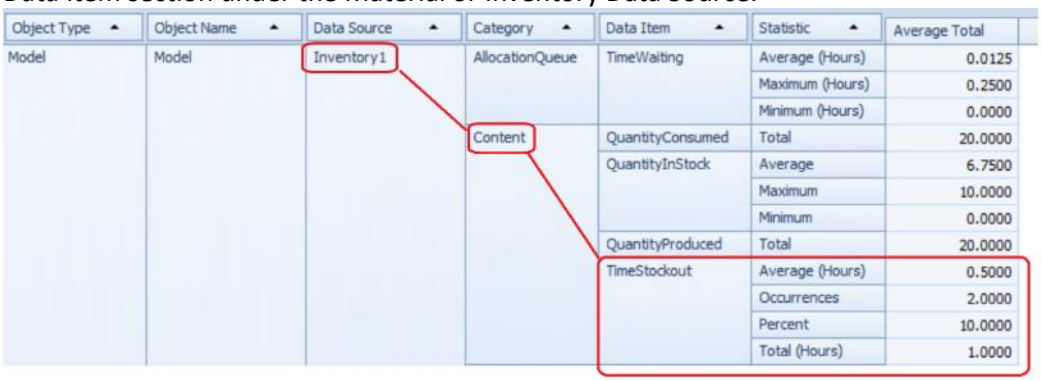
Material and Inventory Stockout Functions and Statistics

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
					Minimum	0.0000
				QuantityProduced	Total	20.0000
				TimeStockout	Average (Hours)	0.5000
					Occurrences	2.0000
					Percent	10.0000
					Total (Hours)	1.0000

Material and Inventory Fill Rate / Demand / Balked / 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 reneging 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 reneged 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 reneged 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	Inventory1	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

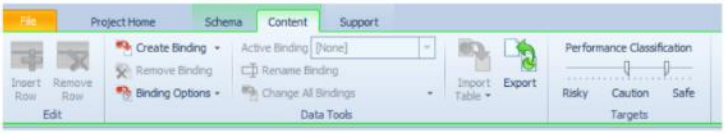
versions). When you first navigate to Data/Tables, if no tables exist yet, Simio defaults to the Schema ribbon. Otherwise, the Content ribbon is shown.

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!

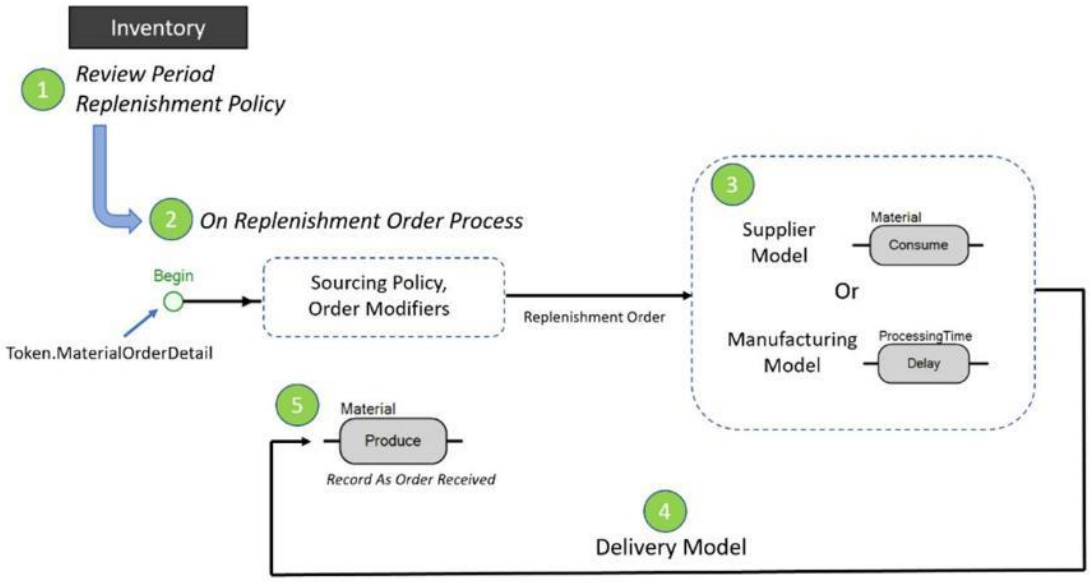
Object and Element Naming Enhancement – RPS Edition
Simio now allows dash and other characters within object and element names. For example, the name of a entity type may be T0-3456, where in prior versions of Simio, the user would have to modify this name slightly to T0_3456 as the '-'dash character 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 ≤ 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.

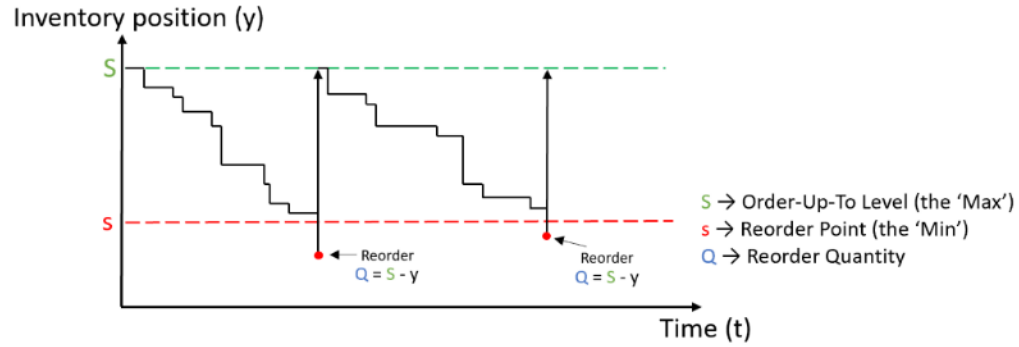


Figure 1 – Continuous Review Min/Max Replenishment Policy

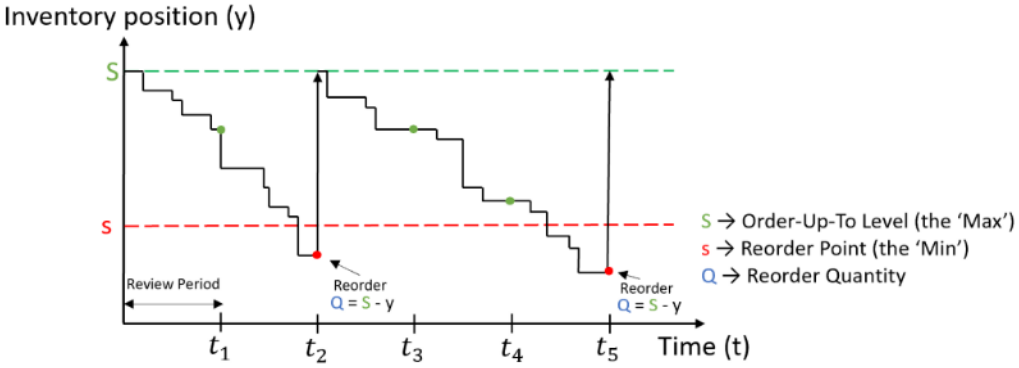


Figure 2 – Periodic Review Min/Max Replenishment Policy

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.

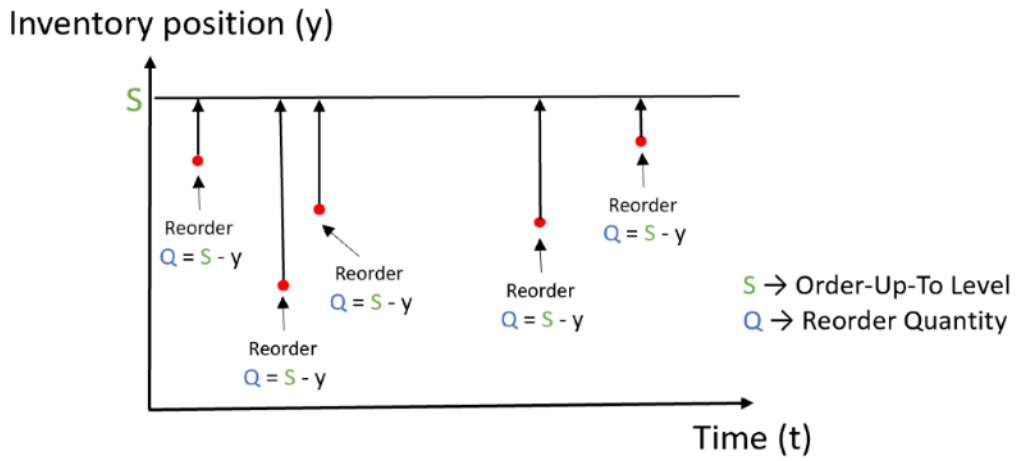


Figure 3 - Continuous Review Order-Up-To Replenishment Policy

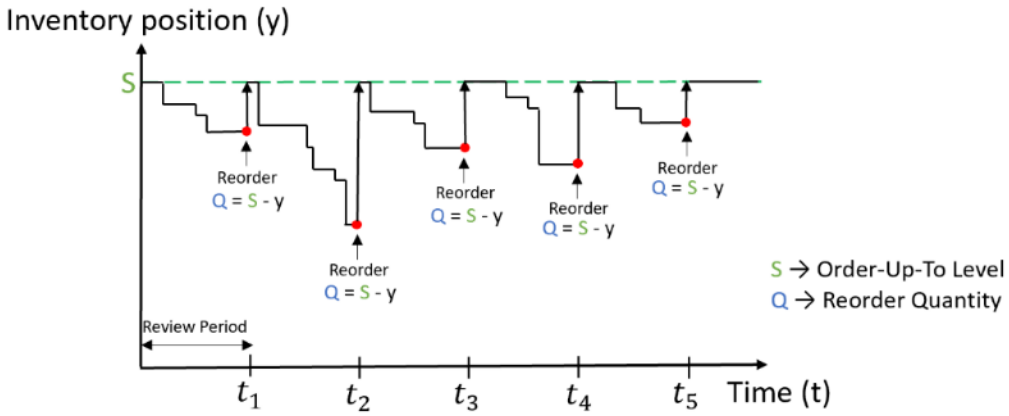


Figure 4 - Periodic Review Order-Up-To Replenishment Policy

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 ≤ s, then a replenishment order of size nQ is required, where n is the smallest integer such that y + *n*Q > s.

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

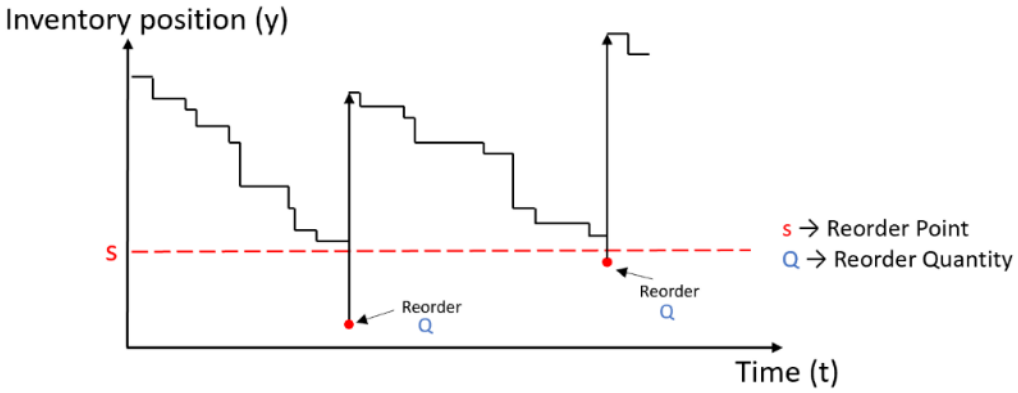


Figure 5 - Continuous Review Reorder Point/Reorder Qty Replenishment Policy

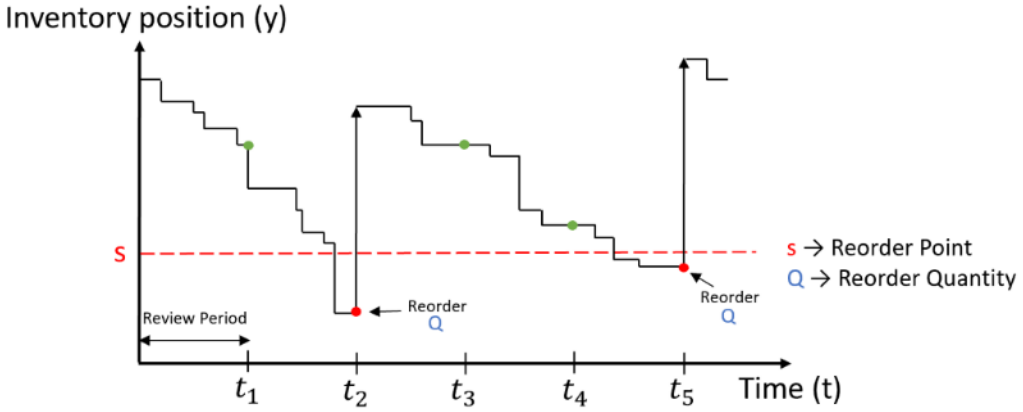
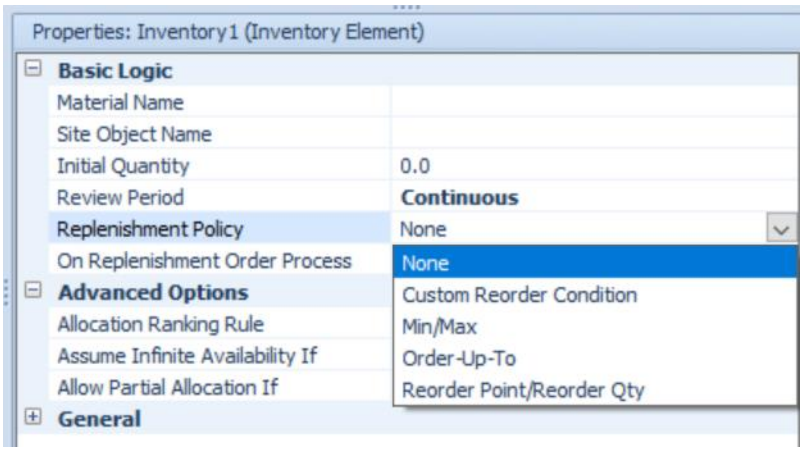


Figure 6 - Periodic Review Reorder Point/Reorder Qty Replenishment Policy

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

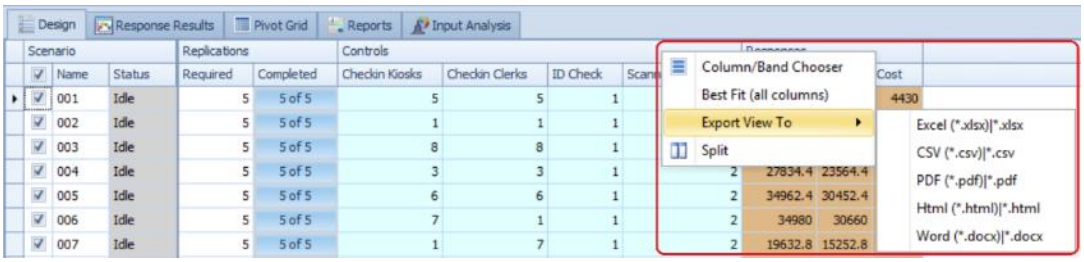
We have added a new SimBit to illustrate the new inventory type features that were added to the previous Sprint 176. Additionally, we have modified the DBReadWrite SimBit to use SQL Server Express.

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.



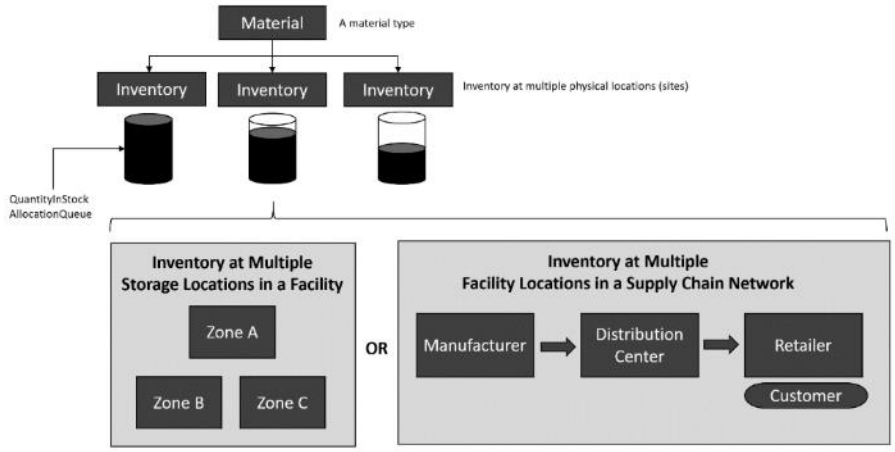
Simio Release 10 – Sprint 175/176 – August 14, 2018

.

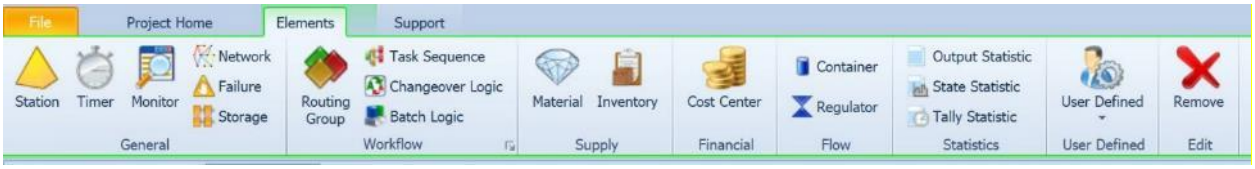
This sprint we have added a new Inventory element that makes it easier for users to model material stock that is stored at multiple physical locations in the system, with each inventory having its own stock level and allocation queue. Now, a fixed object within the Facility window (such as Server or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

New Inventory Element

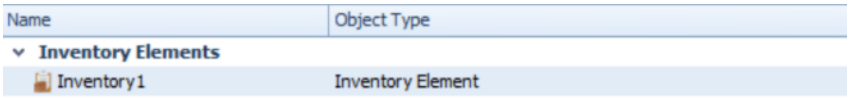
A new Inventory element may be used to define a storage bucket for holding stock of a specific material. A *Site Object Name* property is provided to designate a fixed object in the Facility window, such as a Server or TransferNode, as the inventory's physical location.

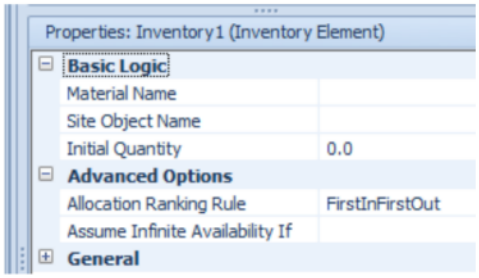


The Inventory element is now available from the Elements ribbon. *Note: Organization of buttons within the Elements ribbon has also changed slightly.



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





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.

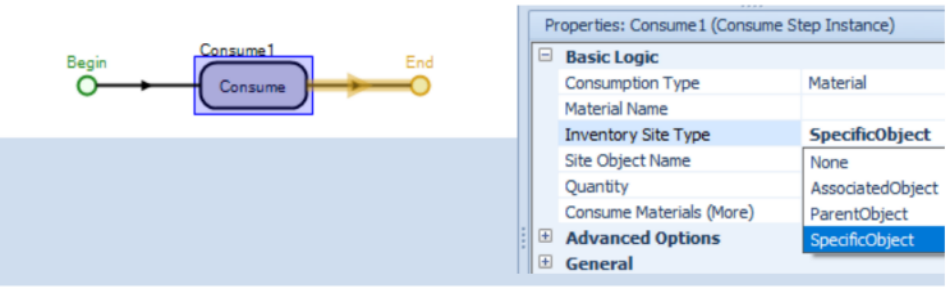
The functions for the Inventory element include:
AverageQuantityInStock - Returns the average quantity of material present in the inventory during the simulation run.
MinimumQuantityInStock - Returns the minimum quantity of material present in the inventory during the simulation run.
MaximumQuantityInStock - 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

Consume, produce, or reserve material 'MaterialA' at 'Site1'.

Material Inventory Lookup

Inventory Name	Material Name	Site Object Name	QuantityIn Stock
Inventory1	MaterialA	Site1	50
Inventory2	MaterialB	Site1	50
Inventory3	MaterialC	Site1	50

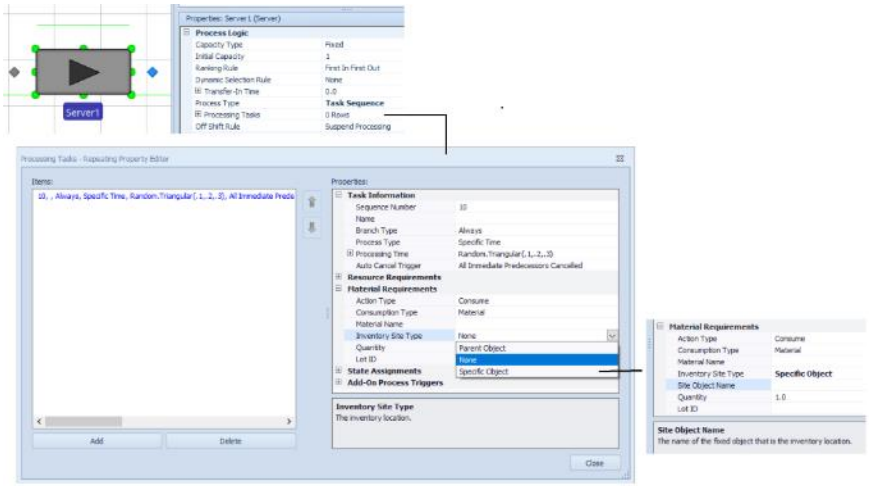
Consume, produce, or reserve bill of materials for 'MaterialA' at 'Site1'.

Component Material Inventory Lookups

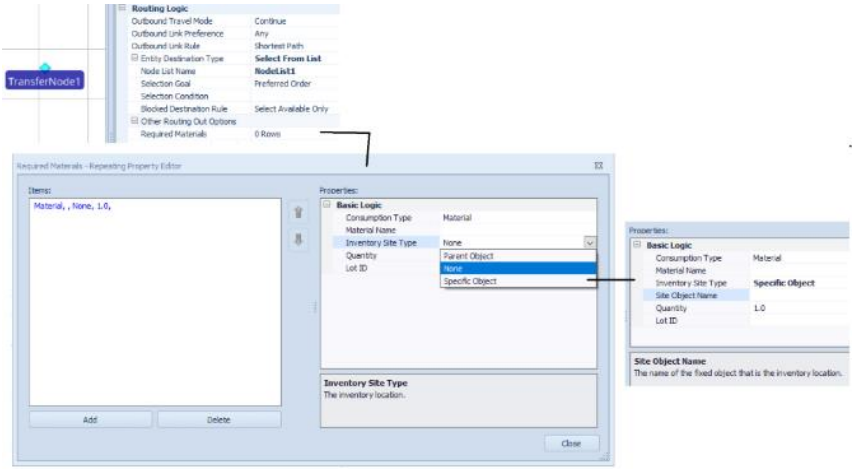
Inventory Name	Material Name	Site Object Name	QuantityIn Stock
Inventory1	MaterialA	Site1	50
Inventory2	MaterialB	Site1	50
Inventory3	MaterialC	Site1	50

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

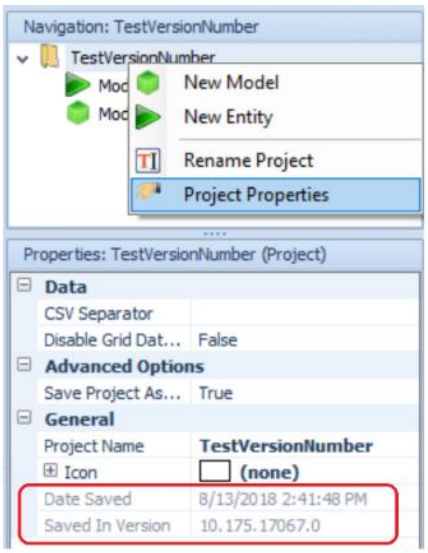
We have added two properties to the Custom Column fields within the Resource Usage Log and the Task Log. The 'Add Column' button on the Log ribbon is used to add custom columns to the logs. This is typically done to provide the user with 'Additional Details' information on a tooltip or to categorize 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
New Model
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
(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

.

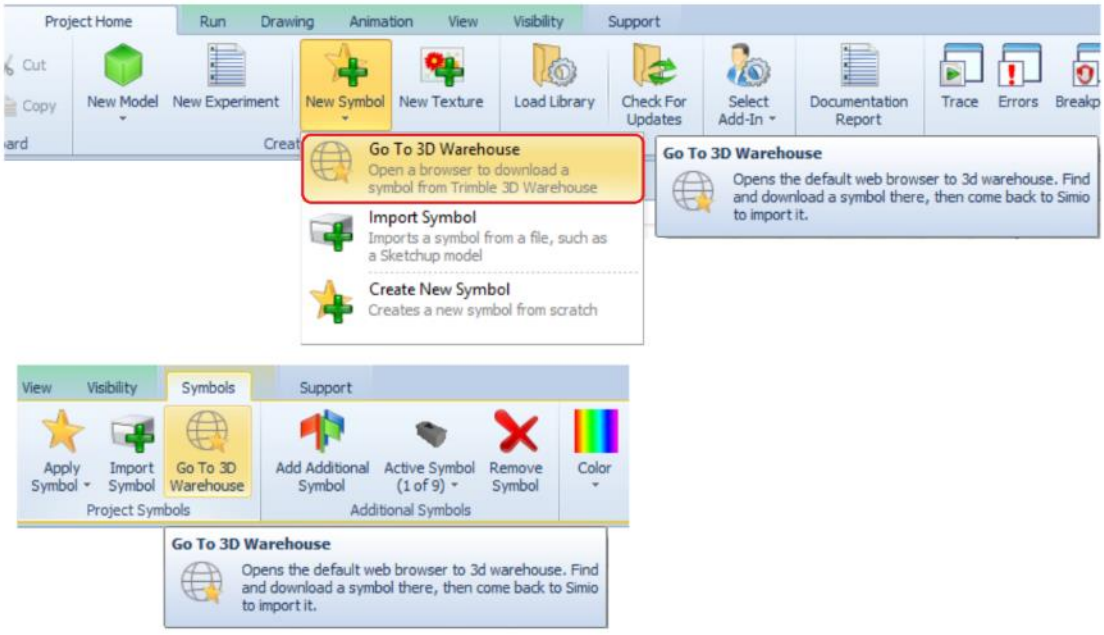
This Simio sprint 174 is our summer public release. In addition to the many features we have added since the last public release sprint 165, this sprint we have modified the 3D Trimble Warehouse interface, enhanced the Gantt charts to include visualization of task constraints in the main Gantts, and 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.



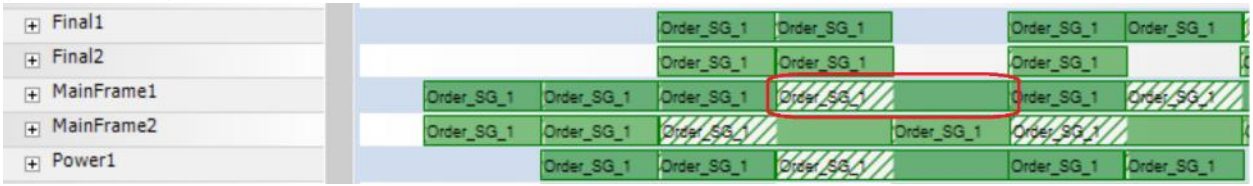
Support for Sketchup 2018

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.

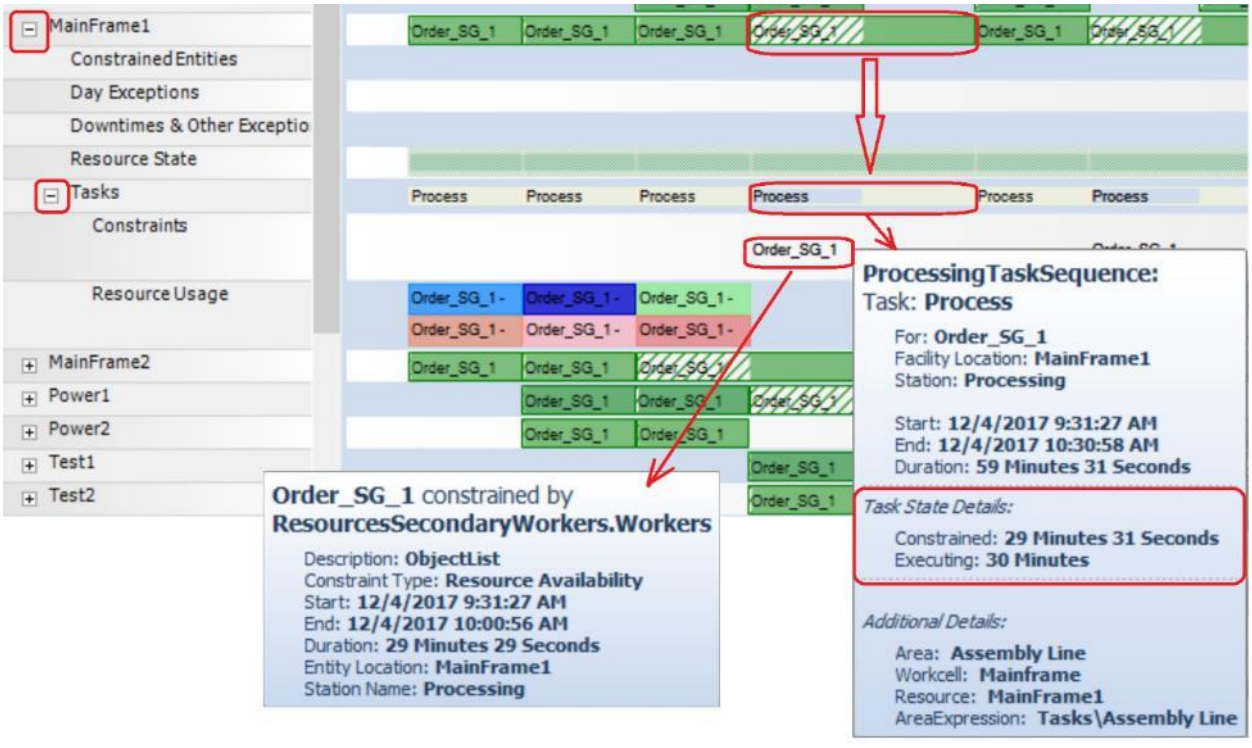
Gantt Enhancements – Enterprise Edition

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).



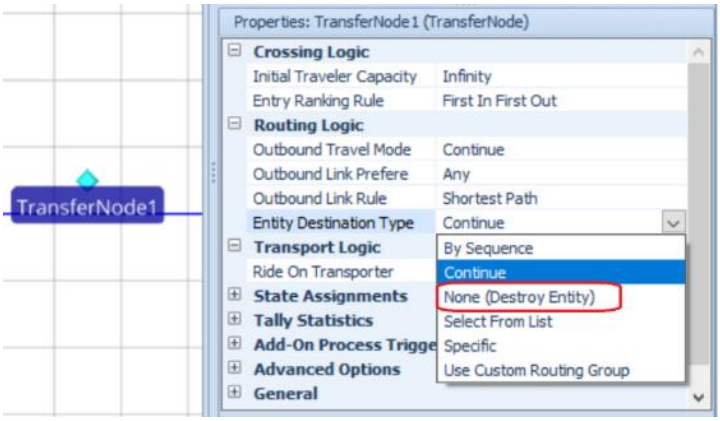
By using the + to open up the MainFrame1 resource, the Process task is shown as constrained. Hovering over the "Process" bar will show the task state details and time of constrained / executing states. Using the + to open the Tasks will show the constraints for the task (resources/materials). Hovering 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.

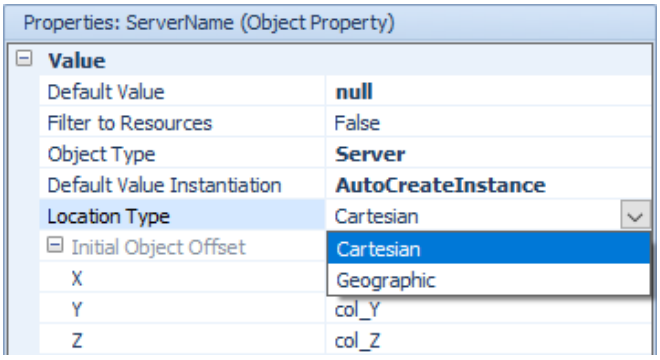
ServersUsingTaskSequenceWithDataTables_LoopbackBranches – This project contains two models in which the operations performed at servers are task sequences, and there are conditional or probabilistic loopback branches in the task workflow. All operation data is defined in a set of relational data tables.

Updated Scheduling Examples

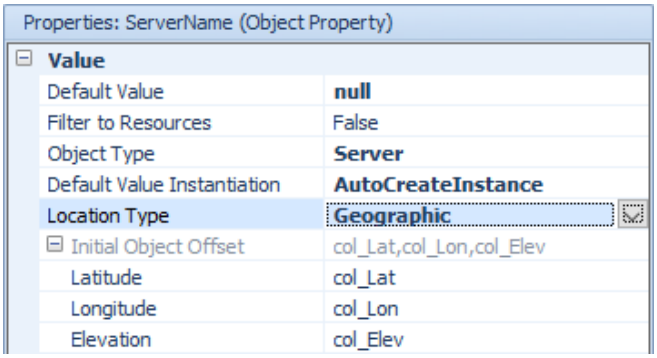
We have updated both the SchedulingBatchBeverageProduction and the SchedulingBicycleAssembly examples to use the auto-create objects functionality within the Resources tables. Note that custom objects are used in conjunction with Simio objects to build these models using the Resource Name column and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



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.

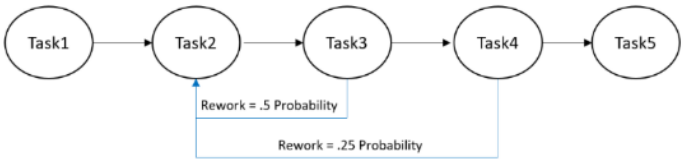
Simio Release 10 – Sprint 172/173 – June 12, 2018

.

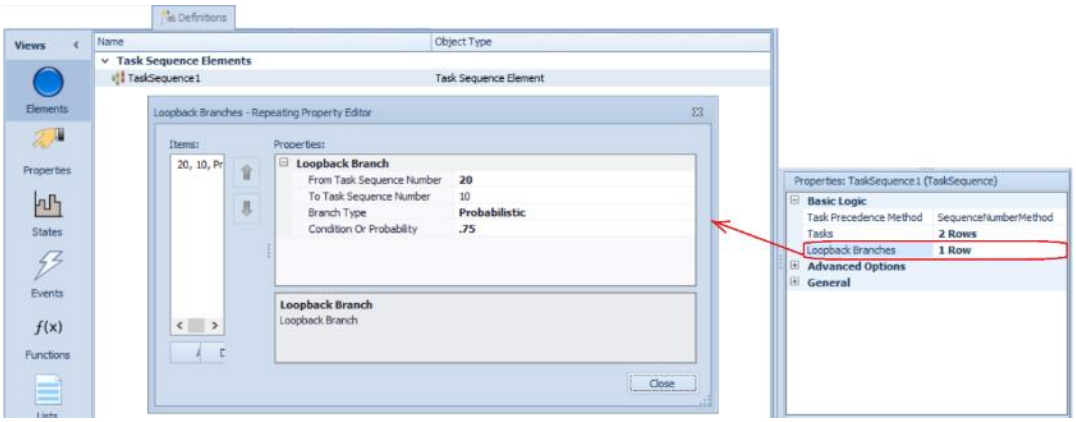
This sprint, we have made several user-requested enhancements to Task Sequences include loopback branches and state assignments within the tasks. We have also fixed some breakpoint issues and changed the way that breakpoints are handled. The Create Object From This feature (added in Sprint 165) has been 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.

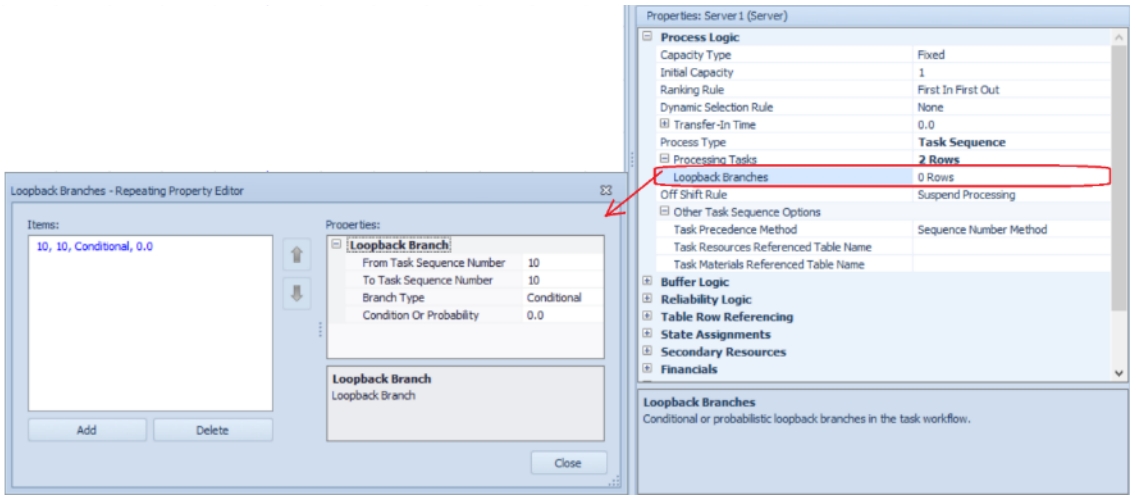


The below diagram shows the required information required for 'SequenceNumberMethod' type sequences. The *From Task ID Number* and *To Task ID Number* properties are used for a *Task Precedence Method* of 'ImmediatePredecessorsMethod' or 'ImmediateSuccessorsMethod'. Multiple loopback branches may be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



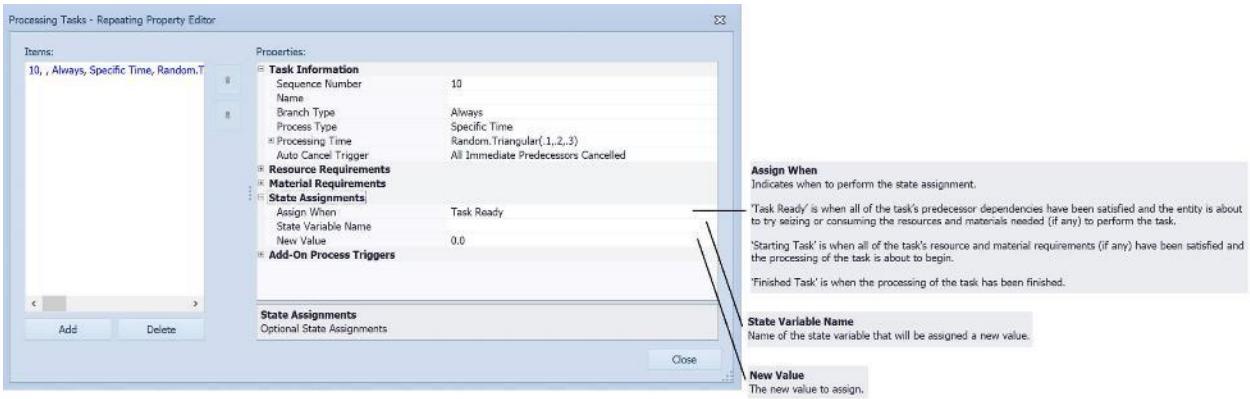
Server, Combiner and Separator – Loopback Branches Enhancements

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.



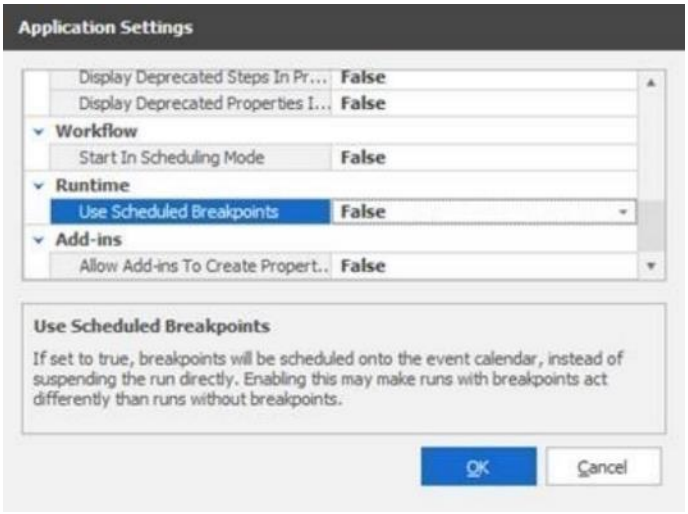
Server, Combiner, and Separator – Task Sequence Process Type Enhancements

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.

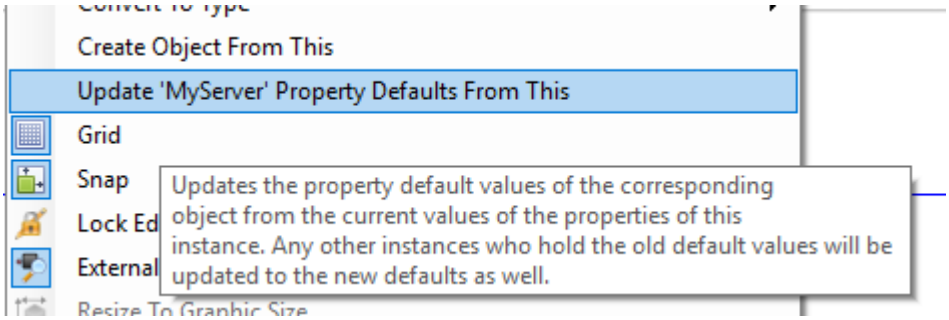
[Assign] Assign1	Assigning state variable 'Model.StringState1' the value 'Fred'. Previous value was ''.
	Assigning state variable 'Model.StringState1' the value 'Barney'. Previous value was 'Fred'.
	Assigning state variable 'Model.ObjectReferenceState1' the value 'Resource1'. Previous value was 'Nothing'.
	Assigning state variable 'Model.ObjectReferenceState1' the value 'Resource2'. Previous value was 'Resource1'.

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 State Log

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.

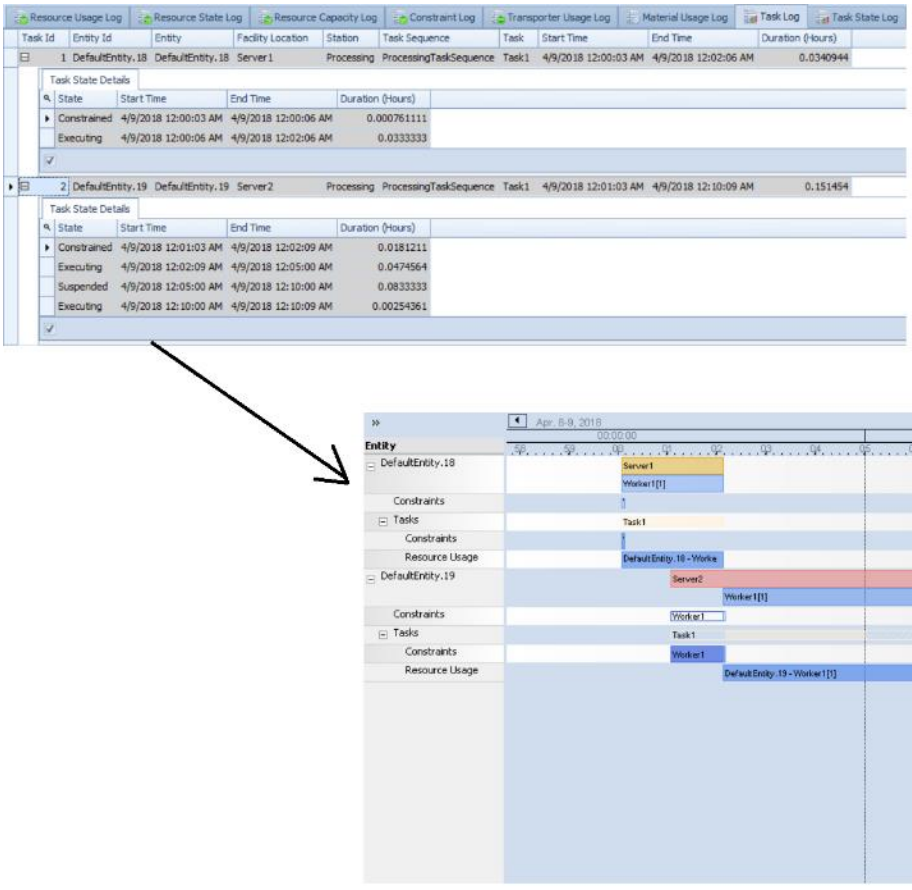
The *Reset Event Name* property is now available if a Timer's *Interval Type* is 'ArrivalTable'. Resetting a Timer that is using an arrival table to schedule events simply resets its EventCount (may be pertinent if using the Maximum Events stopping condition) and resets its ElapsedTime state value.

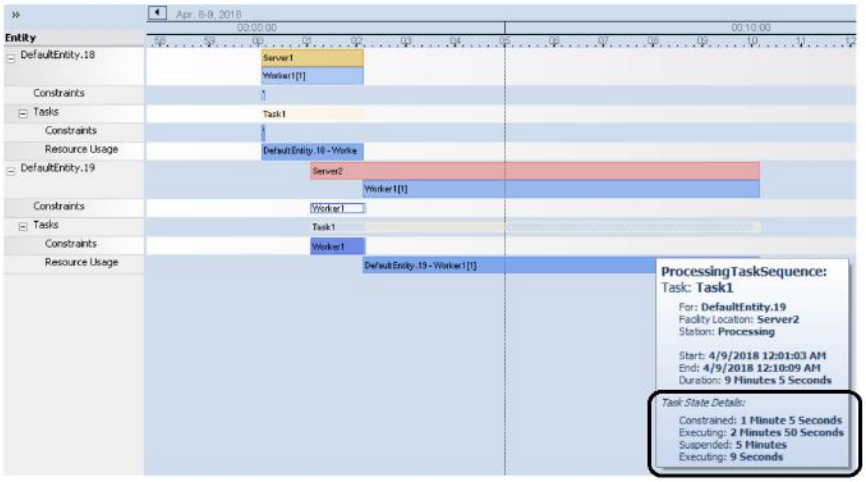
If the *Repeat Arrival Pattern* property is 'True', the Timer now re-evaluates the *Arrival Events Per Time Slot* property at each discrete time in the table on every recurrence. Thus, you can now reasonably specify a random distribution for the number of arrival events per time slot. Previously, that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


Resource Usage Log
Resource State Log
Resource Capacity Log
Constraint Log
Transporter Usage Log
Material Usage Log
Task Log
Task State Log
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 | 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 | 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


Entity
Apr 8-9, 2018
00:00:00
00:10:00
DefaultEntity.18
Server1
Worker1[1]
Constraints
Tasks
Task1
Constraints
Resource Usage
DefaultEntity.18 - Worke
DefaultEntity.19
Server2
Worker1[1]
Constraints
Worker1
Tasks
Task1
Constraints
Worker1
Resource Usage
DefaultEntity.19 - Worker1[1]
ProcessingTaskSequence:
Task: Task1
For: DefaultEntity.19
Facility Location: Server2
Station: Processing
Start: 4/9/2018 12:01:03 AM
End: 4/9/2018 12:10:09 AM
Duration: 9 Minutes 5 Seconds
Task State Details:
Constrained: 1 Minute 5 Seconds
Executing: 2 Minutes 50 Seconds
Suspended: 5 Minutes
Executing: 9 Seconds

Simio Release 10 – Sprint 169 – March 19, 2018

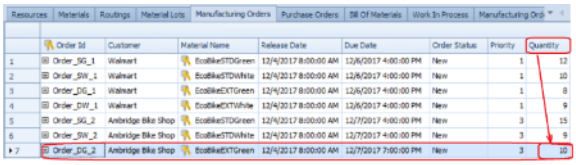
This sprint, we have added several user-requested enhancements, such as a new SimBit for demonstrating worker efficiencies with groups of workers, as well as further categorization for Entity Workflow Gantt charts. Our development staff is also working on many enhancements with our Simio Portal edition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

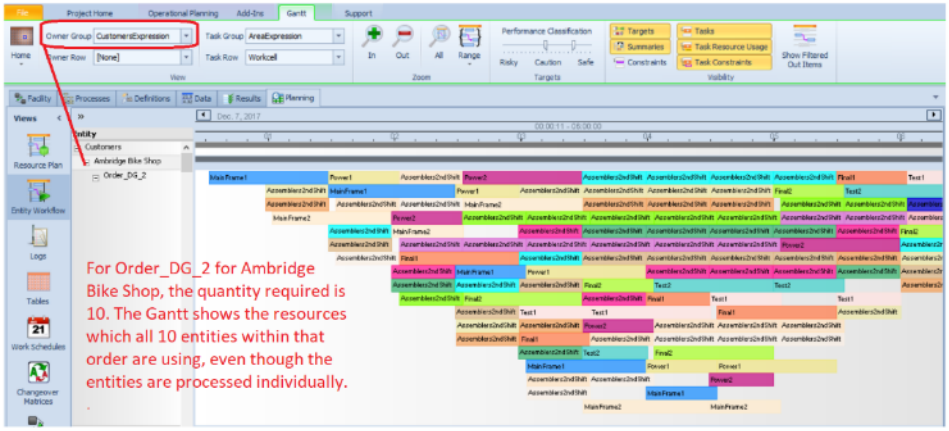
Show Commonly Used Properties Only feature – Removed/Obsolete
Based on customer feedback, we have removed the *Show Commonly Used Properties Only* checkbox that used to reside at the top of all objects. This feature, if checked (default value) would cause only a few of the properties of an object to 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.

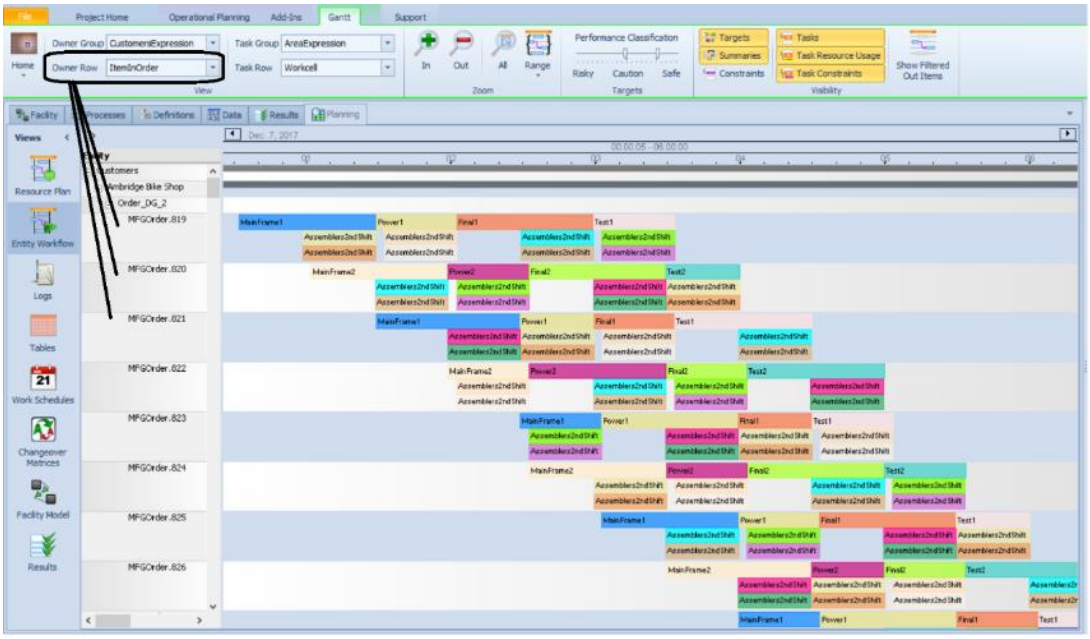
New SimBit
We have added a new SimBit that includes two separate models to demonstrate the use of worker groups and their efficiencies and skills. The project is named WorkerPoolWithEfficiency and includes the below models:
SimpleWorkerPool – Demonstrates how to incorporate individual worker walking 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.





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.

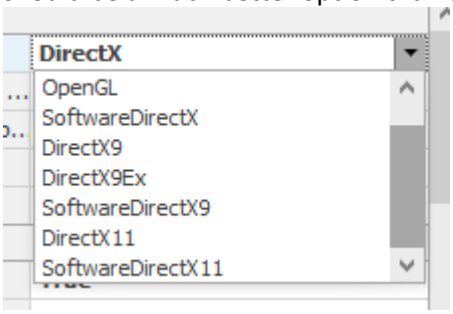
Decision/OnRunEnding Processes - Remove Step Type Restrictions with Continued Error If Any Token Delay

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.

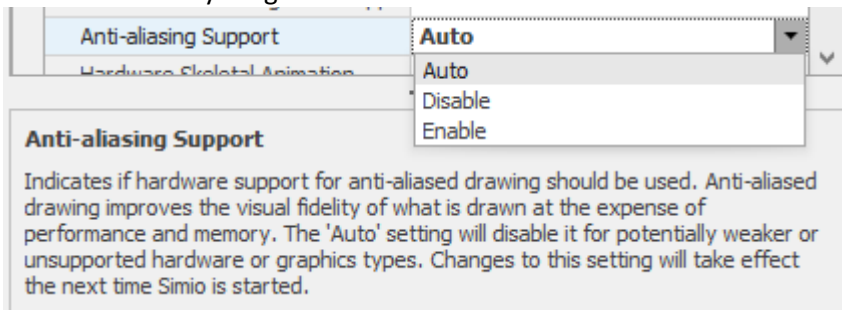
Simio File > Settings … Graphics - DirectX 11, Anti-aliasing, and Hardware Skeletal Animation Support

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), which should be a much better option than the existing SoftwareDirectX9 for VM.



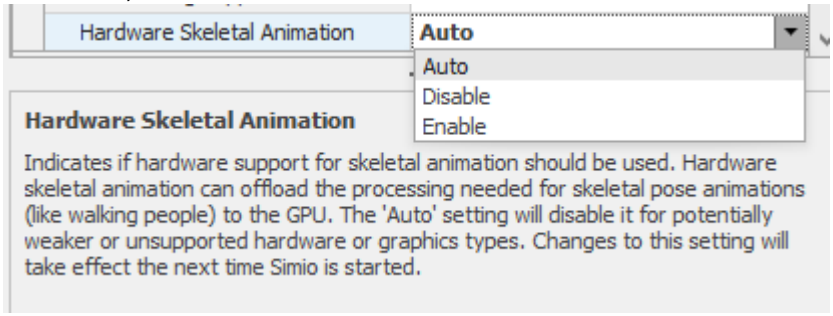
2. A new *Anti-aliasing Support* option. When 'Auto' is selected, will be enabled for DirectX 11 and disabled for everything else.



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.



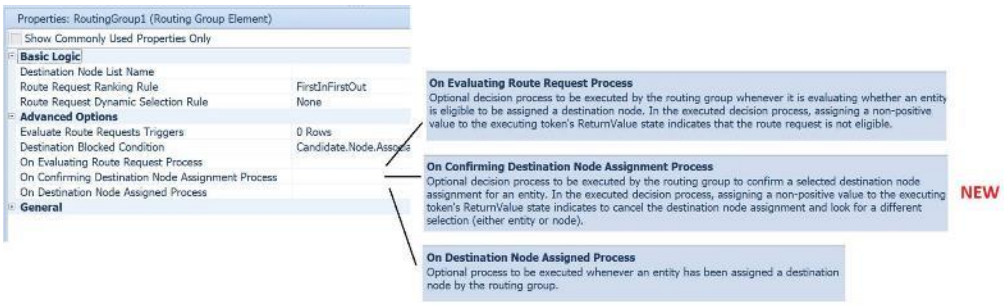
Simio Release 10 – Sprint 167 – February 2, 2018

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).



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.

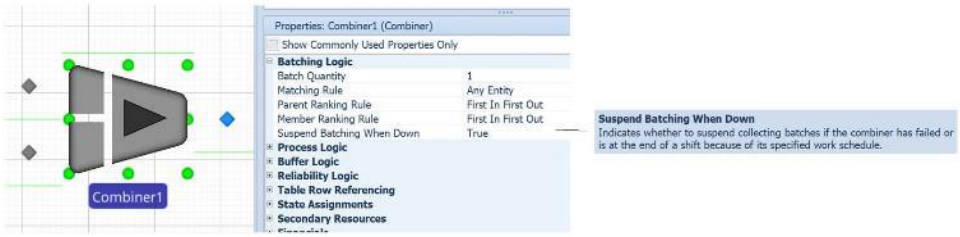
Suspend / Resume Steps – Enhancement

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 reneging 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 reneging 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/reneging 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.

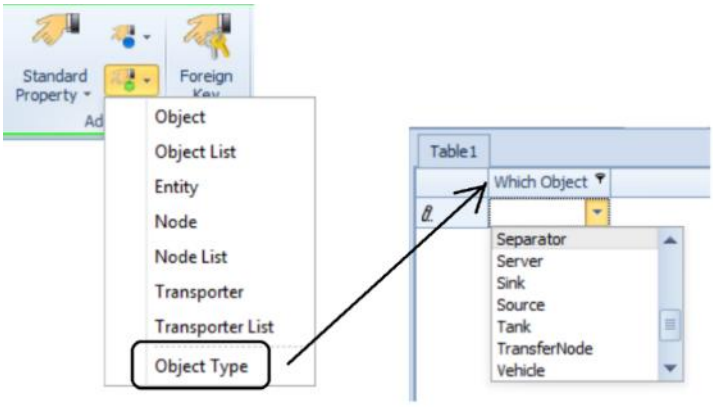
Simio Release 10 – Sprint 164/165 – November 20, 2017

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.

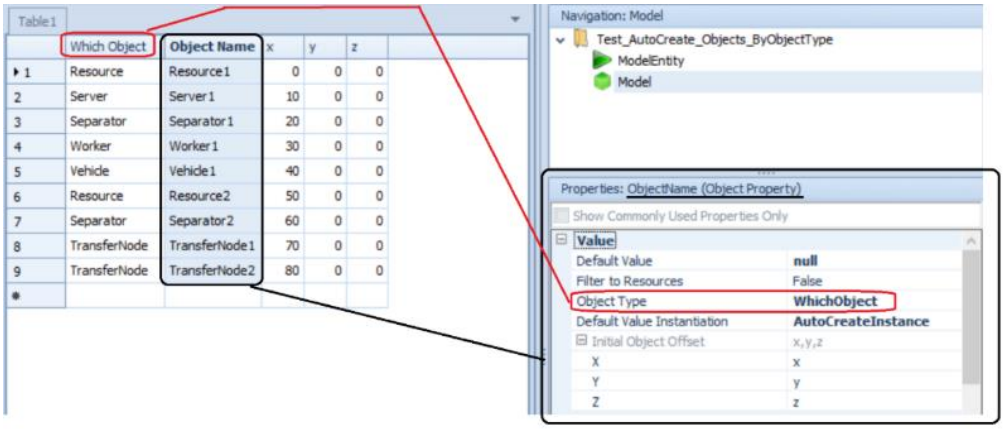
We've also added the ability to right click on an object and change the type of object – let's say from a Resource to a Worker. Within repeating group properties, users can now define 'default tuples' which are default values for one+ repeating property tuple. All these features, along with a right-click 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.



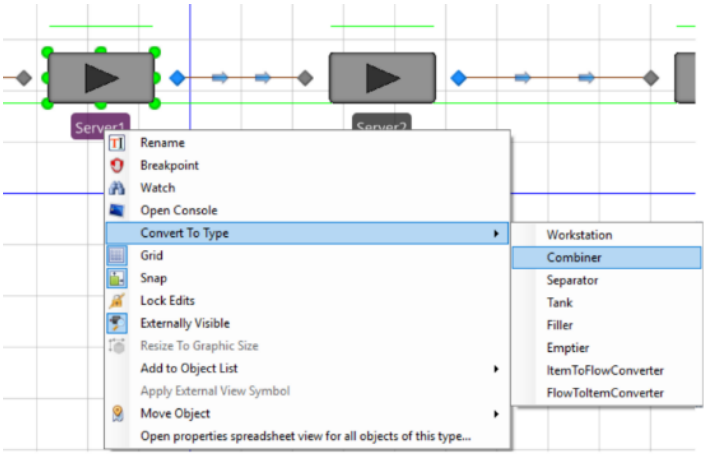
Then, a column with object names that is set to 'auto-create' objects within the Facility window (such as 'ObjectName' column below) can reference the Object Type column (such as 'WhichObject' column below) for the type of object to create.



'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).



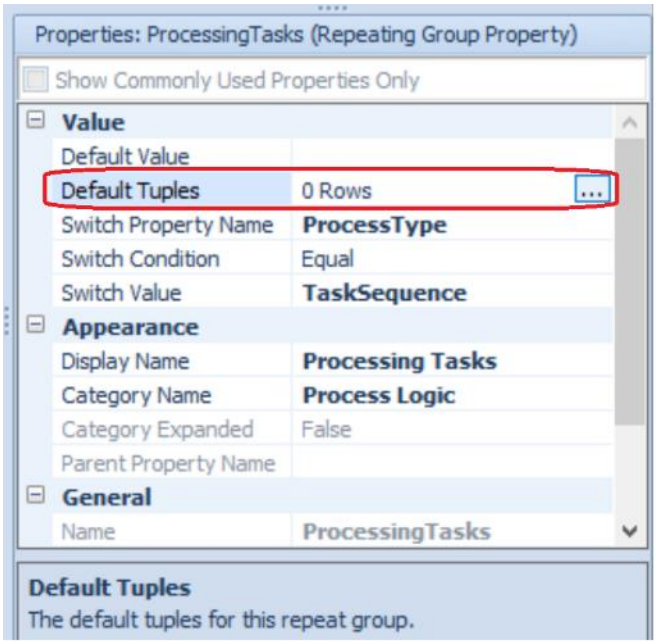
Note that the property value copying works as it currently does, which is to go to the first common shared base class of the two types, and copy those property values from the old instance to the new instance. So, converting between Server -> Combiner will convert all properties of Fixed, but converting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

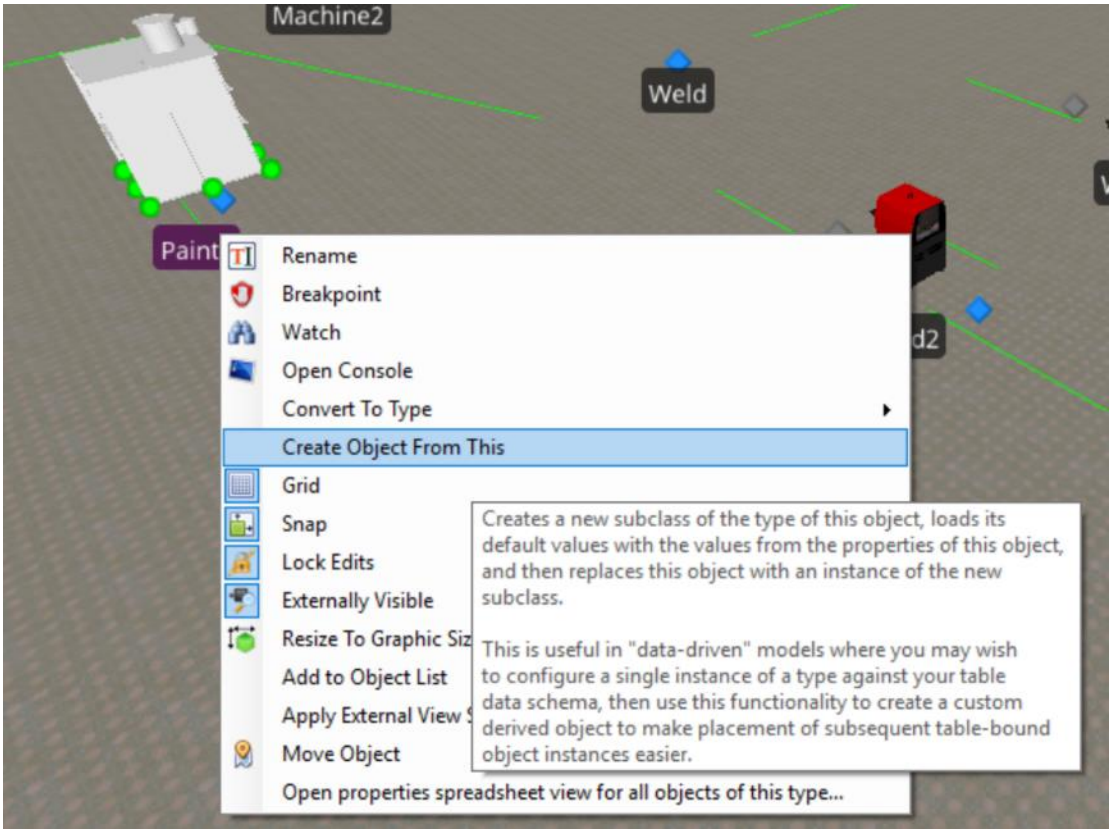
In the example below, a Server was 'sub-classed' and within the properties of the new sub-classed MyServer, the ProcessingTasks property for the Server is a repeating group property. The new *Default Tuples* property can be seen circled in red below. Each *Default Tuples* property, when opened, will 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

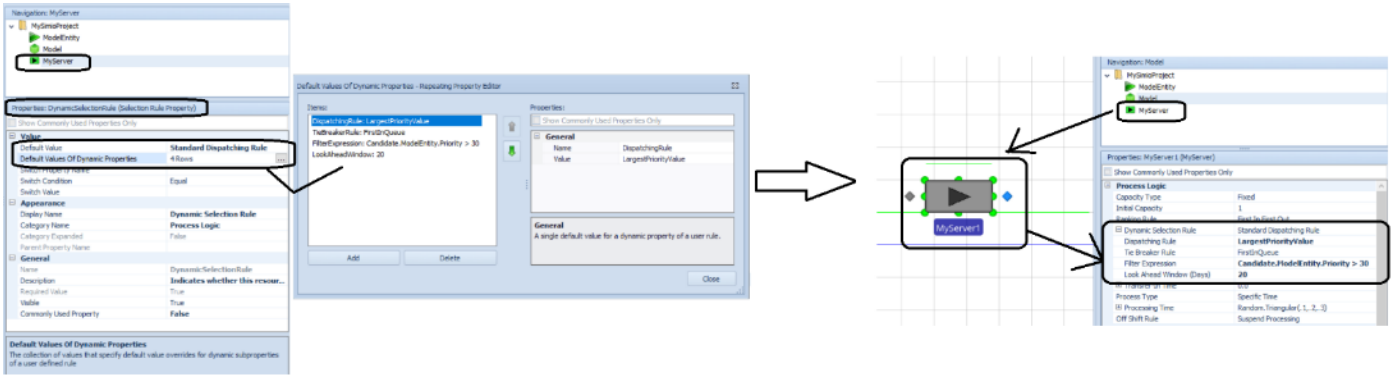
We now have a right click option on objects called 'Create Object From This'. This feature will allow users to enter property information into a Server, for example, including processing tasks, secondary resources, material constraints, capacity/workschedule information, etc. and then right click to 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

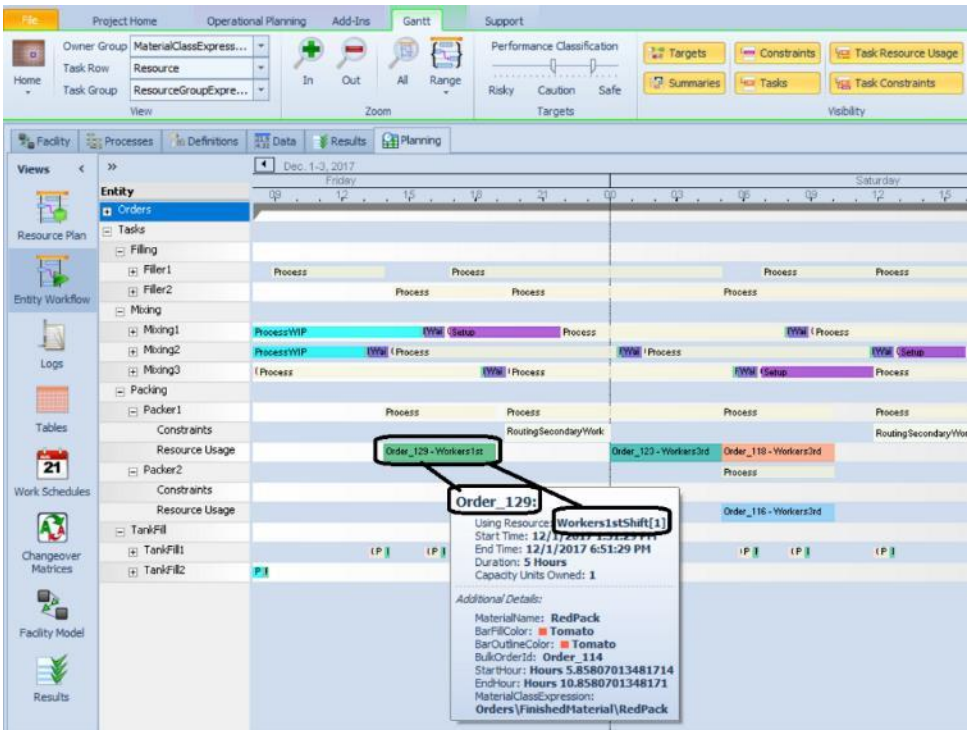
NumberTransferringOut - Returns the current number of entities attempting to transfer out of the station.

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.

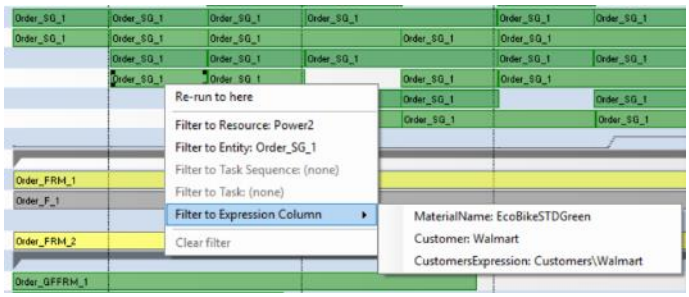
Enterprise Gantt Resource Usage Bars - Concatenate Owner and Resource ID from Resource Usage Log

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!

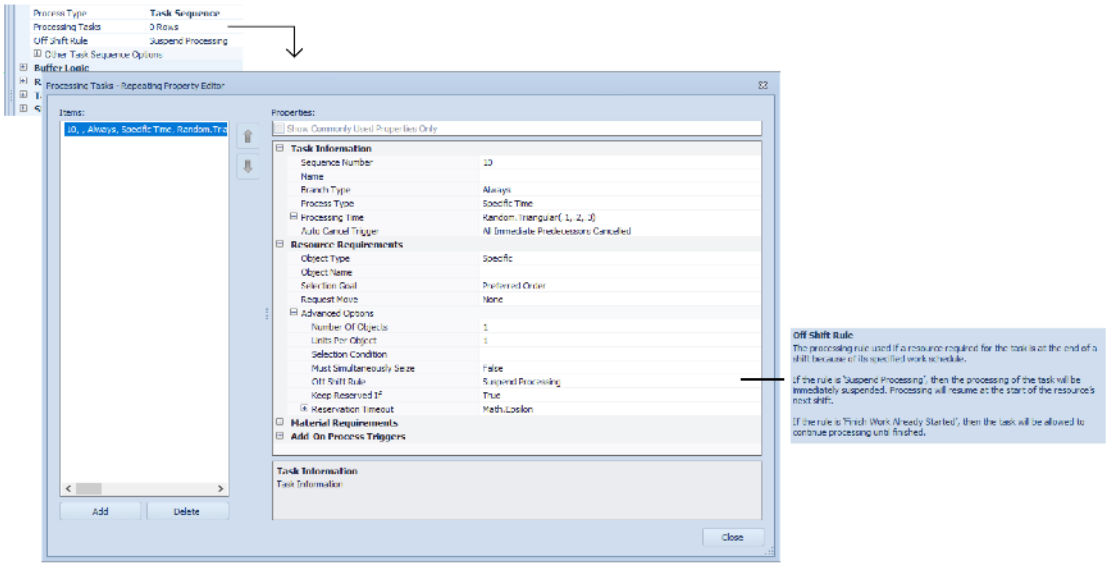
New *Off Shift Rule* for Secondary Resources

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 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.

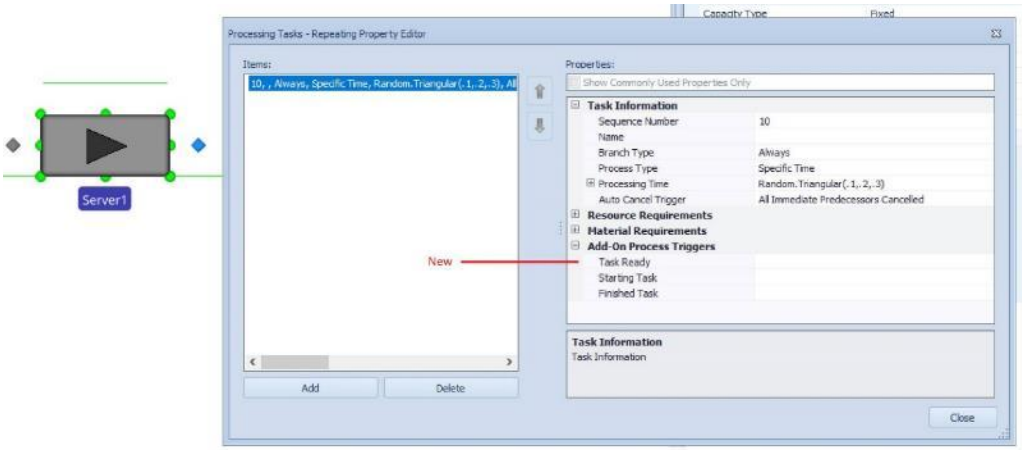
ReservationOwners.Contains (owner) - Returns True (1) if the objects that currently have reservations for capacity units of this resource include the specified object. Otherwise, the value False (0) is returned.

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.

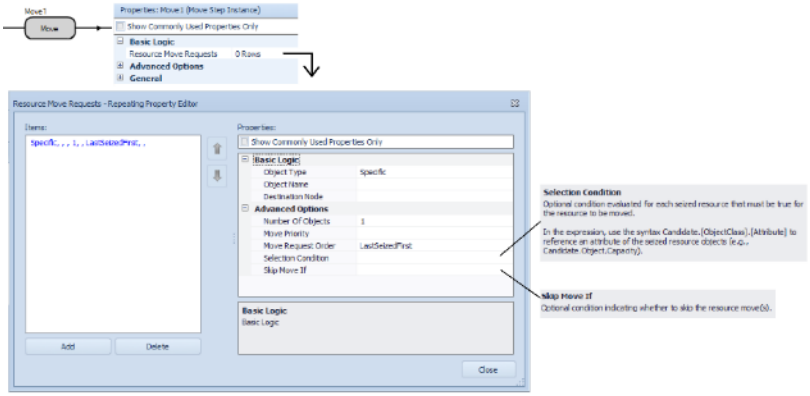
Task Sequences - New *Task Ready* Add-On Process Trigger

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

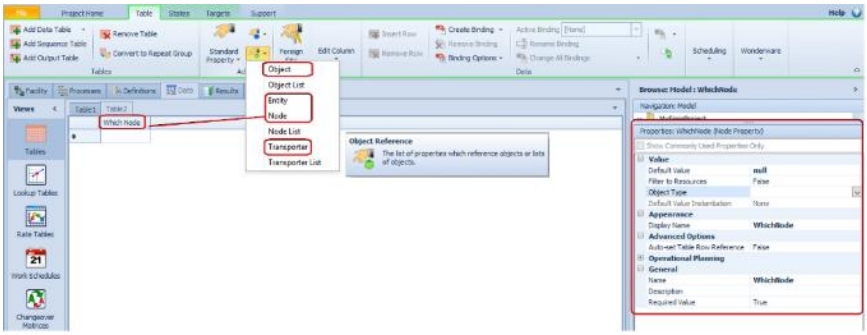
We have added a *Skip Move If* property for the Move step which is an optional condition that will be evaluated for each resource that has a move request. If the expression evaluates to false, the move will not take place. Also within the Move step, we have changed the name of the property *Move Request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



Data Tables Enhancement – Auto-Create for Object Reference Columns

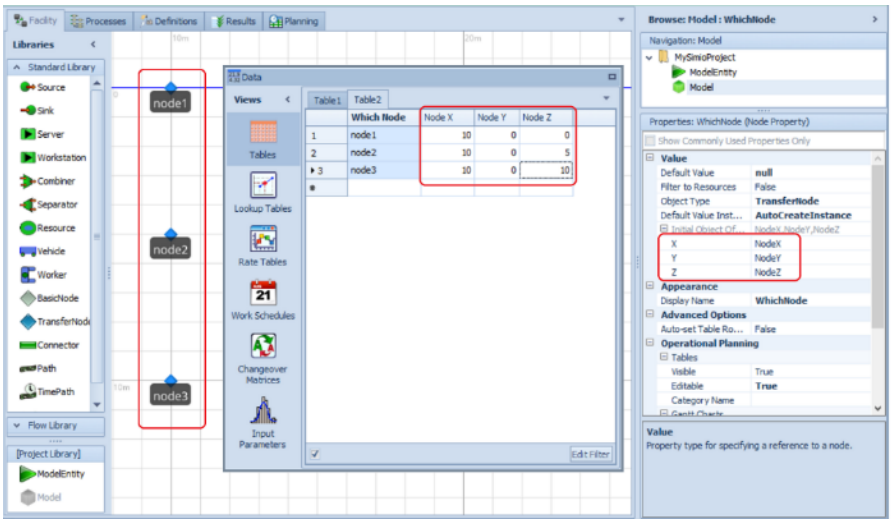
We have added the capability to auto-create objects by using the Object Reference column type within a data table or sequence table. When a column is of type Object Reference (i.e., Object, Entity, Node or Transporter), there are additional properties, including *Object Type*, *Default Value Instantiation*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

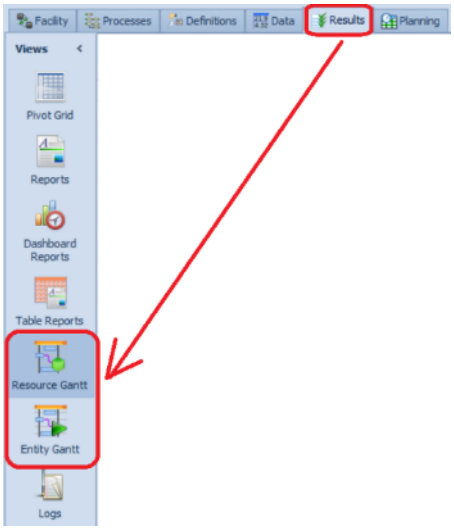
The *Initial Object Offset* values for *X, Y, Z* locations in the Facility window can be specified as the default of 0,0,0 or a real/integer value. These location values can also reference a real/integer column within the same table, where the objects have different location values and are place accordingly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


Facility
Processes
Definitions
Data
Results
Planning
Views
Pivot Grid
Reports
Dashboard Reports
Table Reports
Resource Gantt
Entity Gantt
Logs

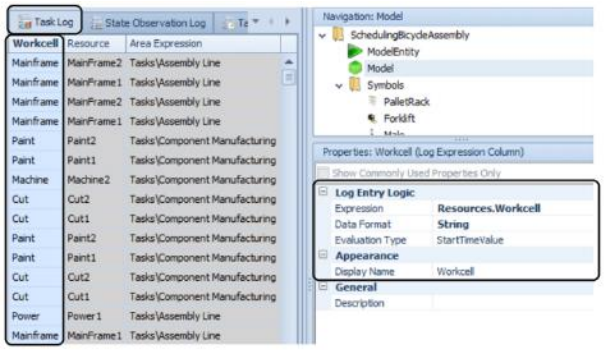
Simio Release 9 – Sprint 162 – September 21, 2017

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.

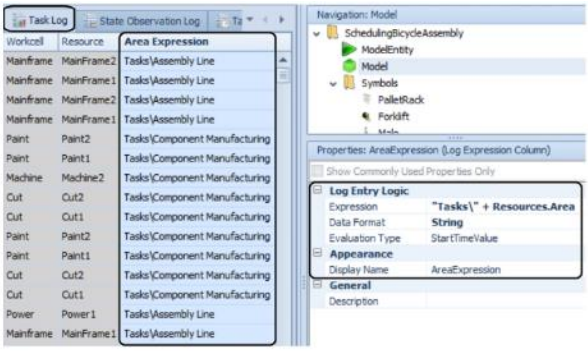
Simio Enterprise Edition – Task

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

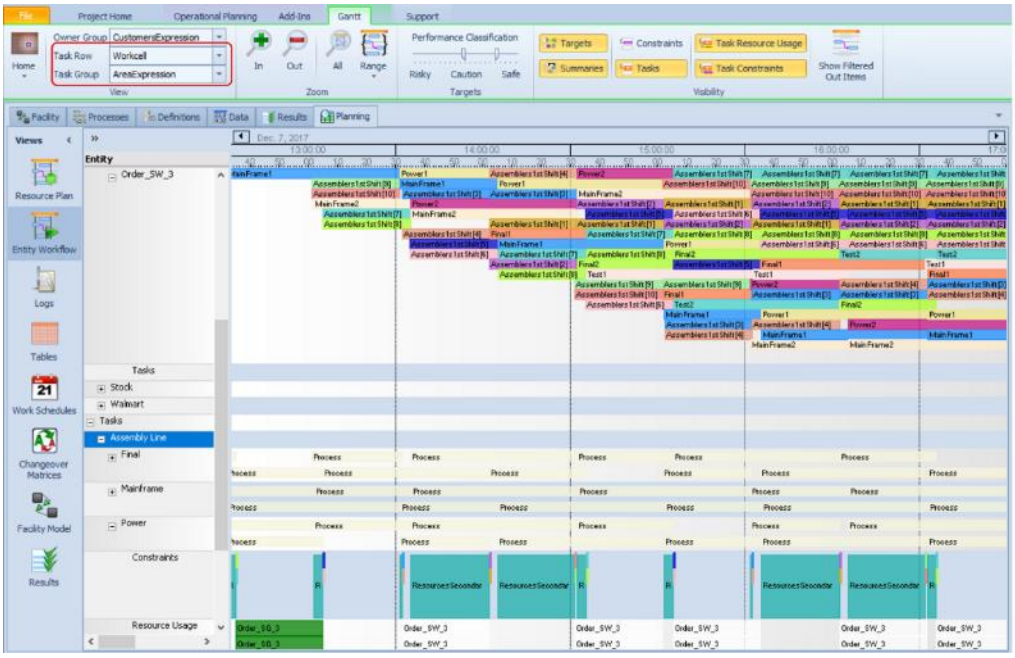


Used for the *Task Group* property

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 Gantts.

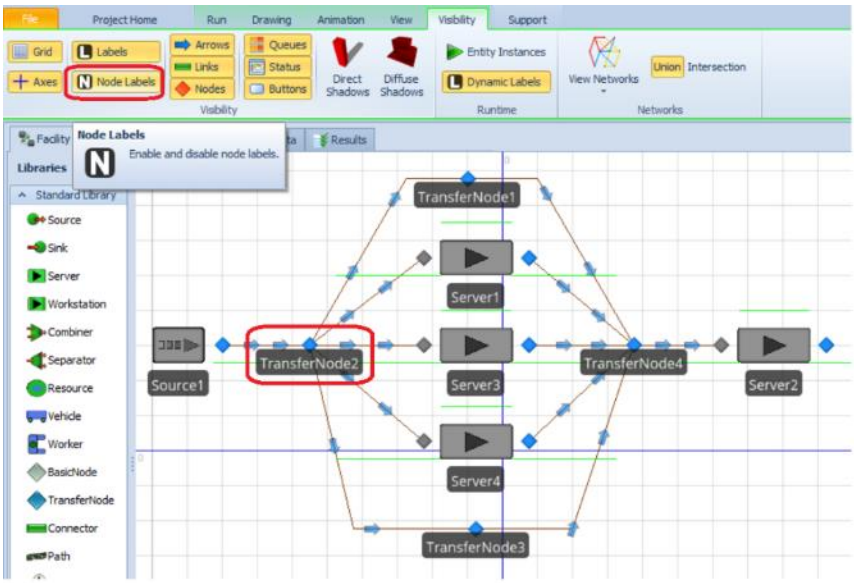


Using the Task Group option for categorizing tasks actually takes the Tasks out from under the particular order number (or entity) and puts them in a Tasks section. This allows the tasks for all orders (entities) to be seen in the same area.

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.

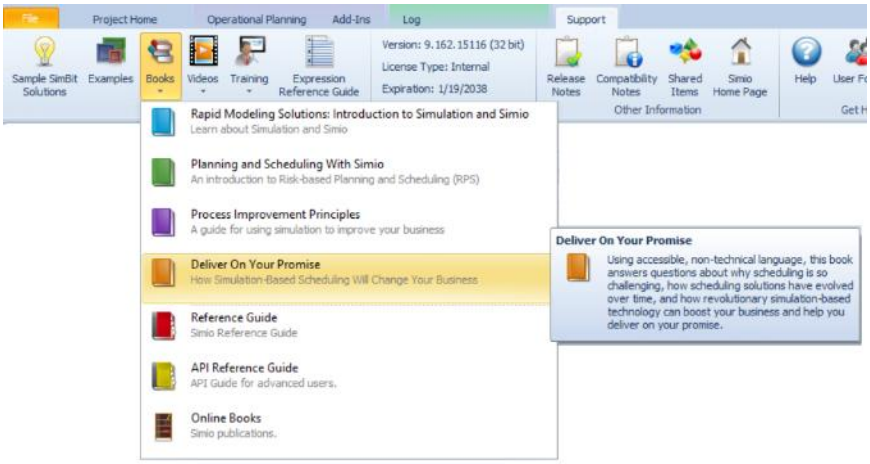
TokensInProcess.NumberItems - Returns the number of tokens that are currently executing the process.
TokensInProcess.FirstItem - Returns a reference to the first token in the list of tokens that are currently executing the process.
TokensInProcess.LastItem - Returns a reference to the last token in the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



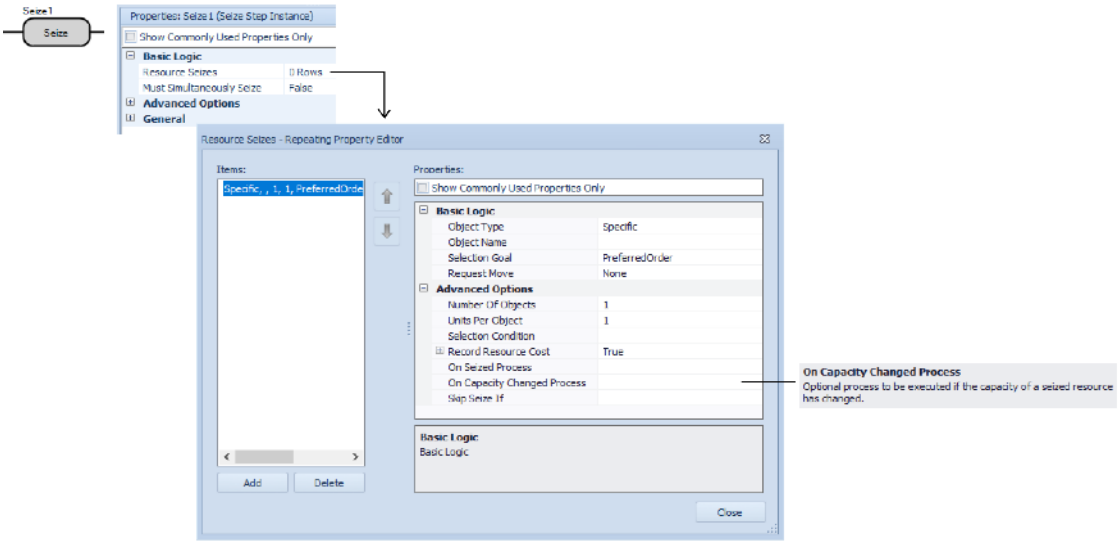
Simio Update to .NET 4.6.2
Simio requires the *.NET Framework Version 4.6.2*. If the framework is not already installed on your computer and you want to perform a manual install, you can download it from the *.NET Framework Downloads* page: (http://msdn2.microsoft.com/en-us/netframework/aa569263.aspx). 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.



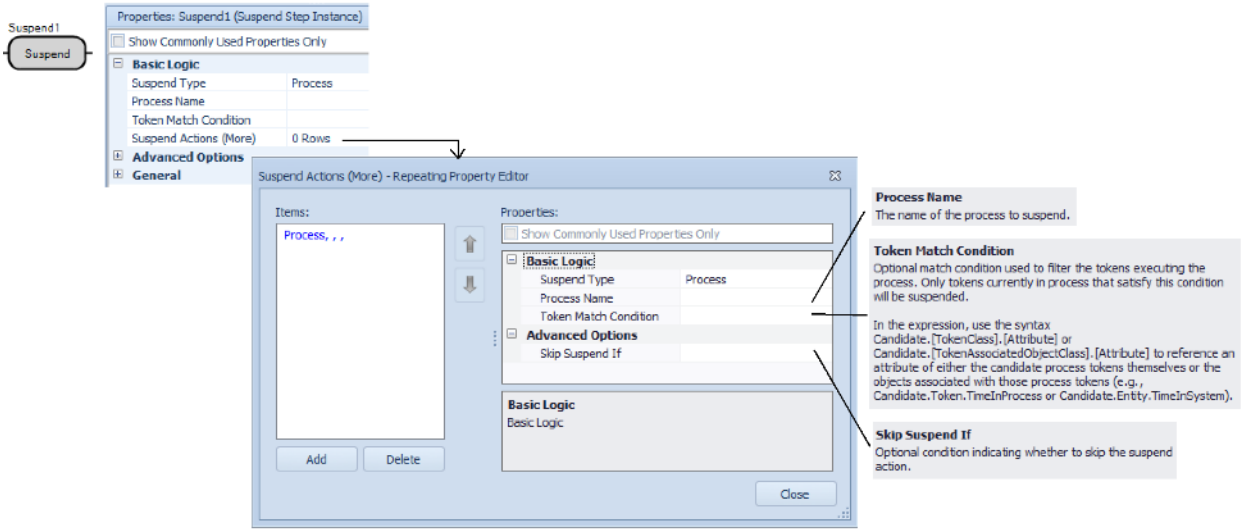
When a seized resource object's capacity has changed, the *On Capacity Changed Process* for each of the resource's owners (if any were specified at the Seize step(s)) will be first executed in seize order. Then last the resource object's *OnCapacityChanged* process.

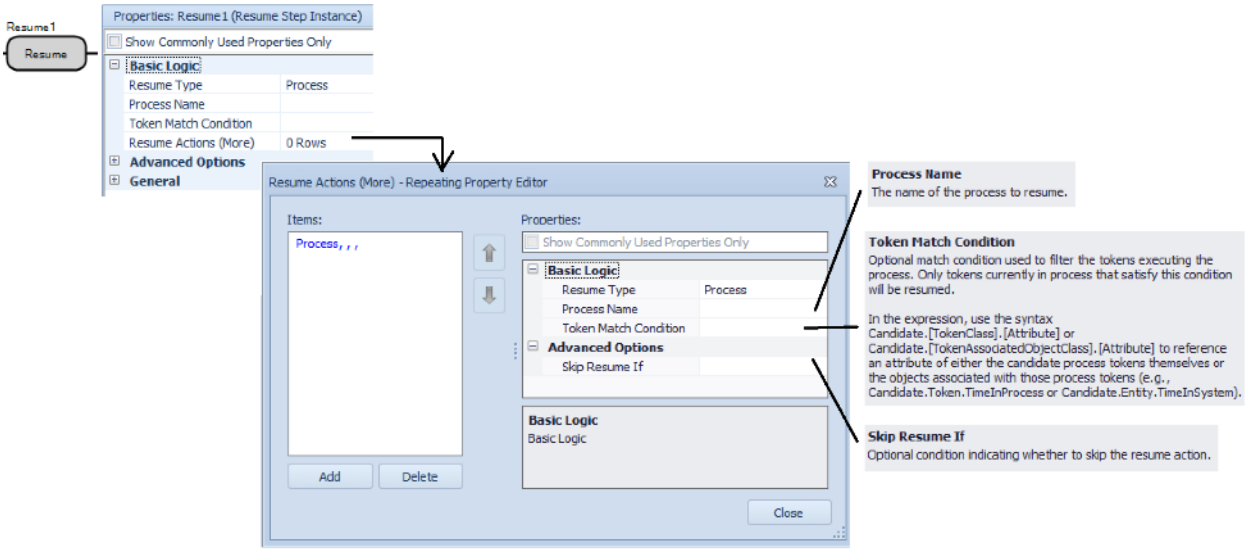
When a new token is created to execute the *On Capacity Changed Process* that was specified at a Seize step:

- 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.

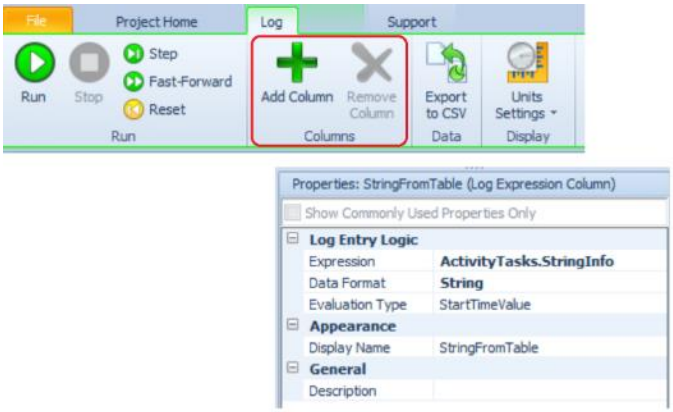
Simio Enterprise Edition – Task Id in Logs, Add Column Expressions within Task Log

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.

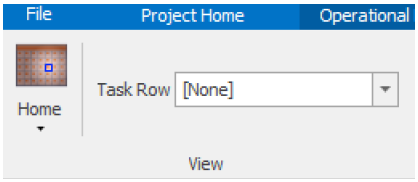
In addition to the Task Id available in the Task Log, we have added the ability to add/remove an expression type column(s) to the Task Log. The Log ribbon includes both the Add Column and Remove Column buttons (similar to adding new expression columns to the Resource Usage Log).



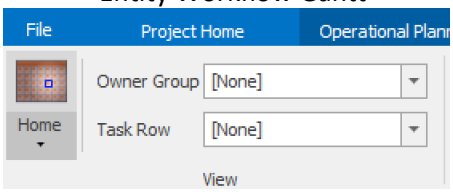
Note that the token used to evaluate the expression is the one created to run the process associated with the task, so whatever expression you would use in the model to get information, you would use here as well (e.g., Entity.** for the associated object that was indicated for StartTask, Token.TaskInfo.**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

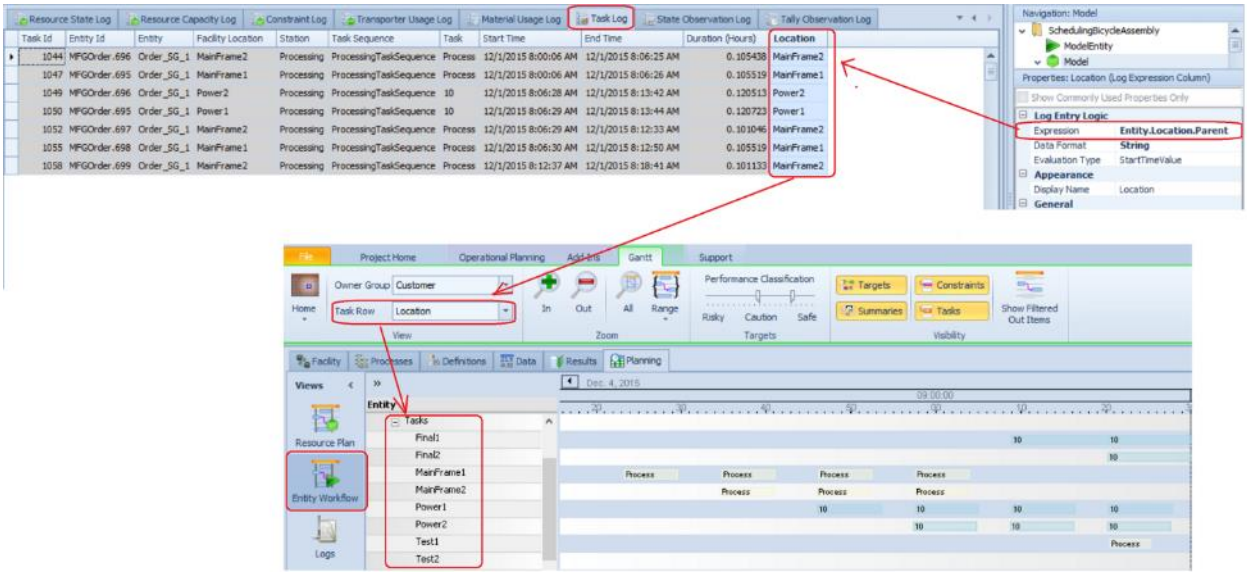
Simio Enterprise Edition – Resource Plan and Entity Workflow Gantts - Task Row combo box selection
We have added a Task Row selection box to both the Resource Plan and Entity Workflow Gantts, where you can specify which custom expression on the Task Log is used to indicate what row a task item goes in. 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.

Resource Gantt



Entity Workflow Gantt



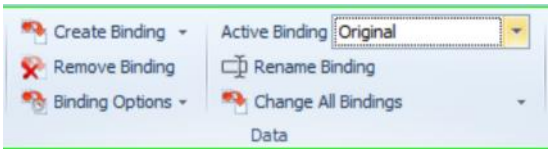

Resource State Log
Resource Capacity Log
Constraint Log
Transporter Usage Log
Material Usage Log
Task Log
State Observation Log
Tally Observation Log
Task Id | Entity Id | Entity | Facility Location | Station | Task Sequence | Task | Start Time | End Time | Duration (Hours) | Location
1044 | MFGOrder.696 | 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.695 | 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 | 10 | 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 | 10 | 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
ModelEntity
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
Home
Owner Group
Customer
Task Row
Location
View
In
Out
All
Range
Zoom
Performance Classification
Risky
Caution
Safe
Targets
Targets
Constraints
Summaries
Tasks
Visibility
Show Filtered Out Items
Facility
Processes
Definitions
Data
Results
Planning
Views
Resource Plan
Entity Workflow
Logs
Entity
Tasks
Final1
Final2
MainFrame1
MainFrame2
Power1
Power2
Test1
Test2
Dec. 4, 2015
09:00:00
Process

Simio Release 9 – Sprint 159 – July 24, 2017

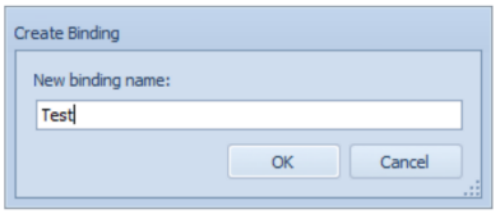
In this sprint, we have made many enhancements to the Data Binding capabilities within data tables. This includes having multiple bindings per table, changing between bindings for multiple tables easily and experiment scenario based data binding. Additionally, we have added the ability to remove rows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

Data Table Binding Enhancements

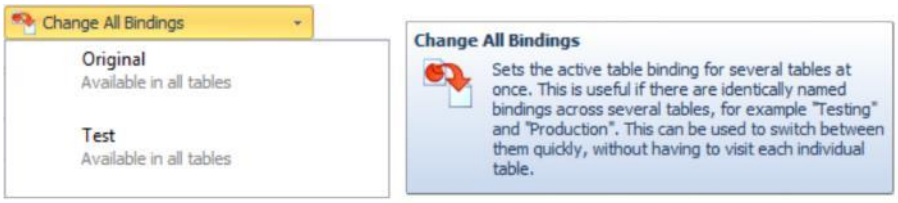
We have enhanced the data binding capabilities for data tables, including adding an *Active Binding* selection to allow for multiple bindings per data table. The original binding for a data table is considered the 'Default' binding (*Active Binding* is 'Default'). The *Rename Binding* option allows users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

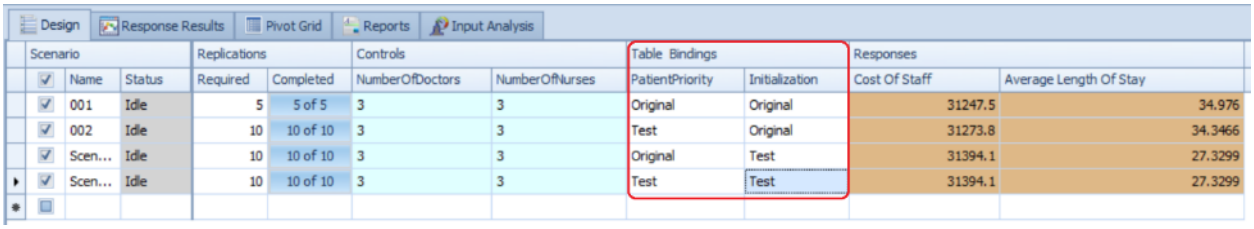


Each table then has its own 'group' of bindings. The *Change All Bindings* option then allows the user to set the Active Binding for several tables at one time. If there are multiple tables with the same binding name, they will be displayed in the *Change All Bindings* pull down list.



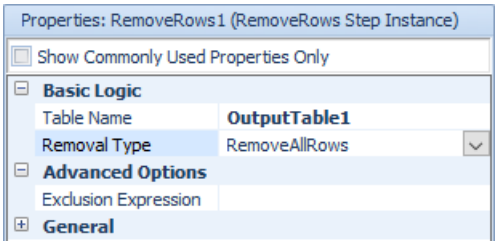
Experimentation – Scenario Based Data Binding

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.

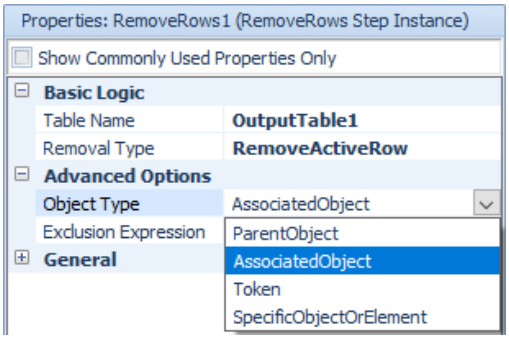


RemoveRows Step Enhancement

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:



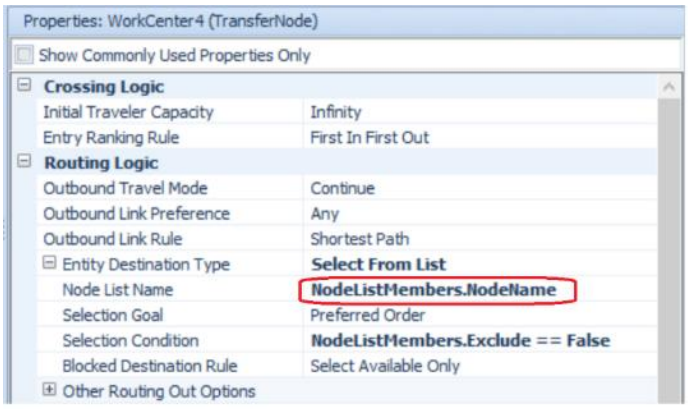
Setting the *Removal Type* to 'RemoveActiveRow' allows users to specify which object or element has a reference to the row to be removed:



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

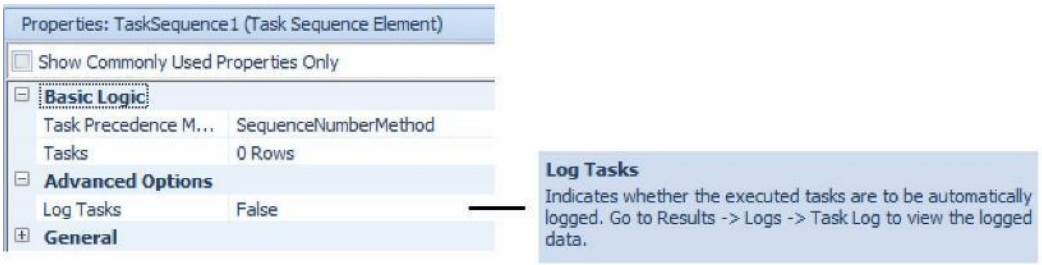
We have added a two new SimBits to demonstrate the use of tables for defining lists of objects.

UsingRelationalTablesToDefineNodeLists – 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.

UsingRelationalTablesToDefineResourceTaskLists – 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)

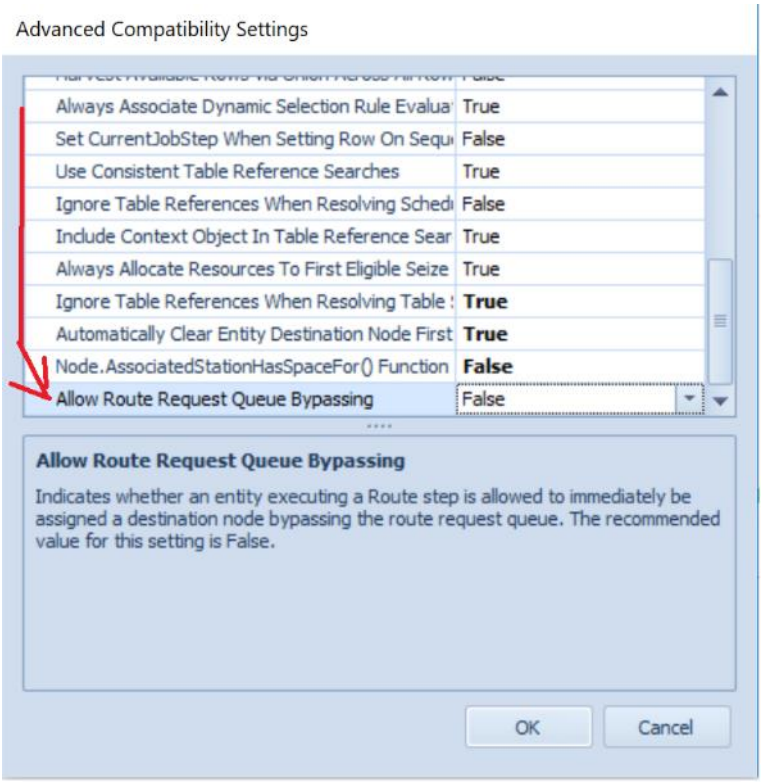
Simio doesn't allow identifiers that evaluate to numbers so '123' is not valid, but '3f4' is valid. '3e4' is not valid because that is exponential notation for 3 x 10^4.

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.

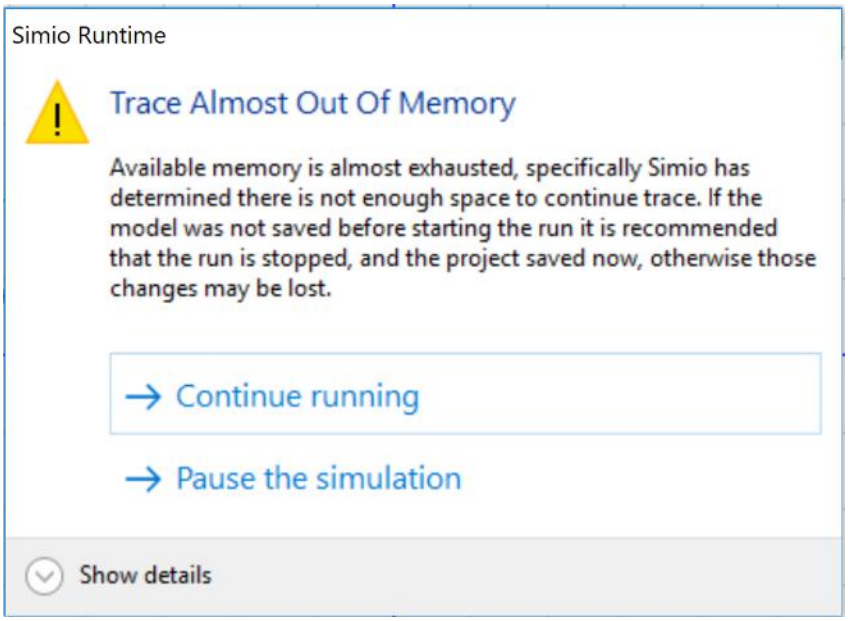


2) When checking its route request queue, the new selection algorithm used by the routing group will be as follows:
   <u>If no dynamic selection rule is being used (static queue ranking rule only)</u>
   - Use the route request queue's static ranking rule to select a queue item.
   - If there are multiple possible destination nodes available for the selected queue item, then use the entity's destination selection goal to select one of those nodes.

   <u>If dynamic selection rule is being used</u>
   - 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

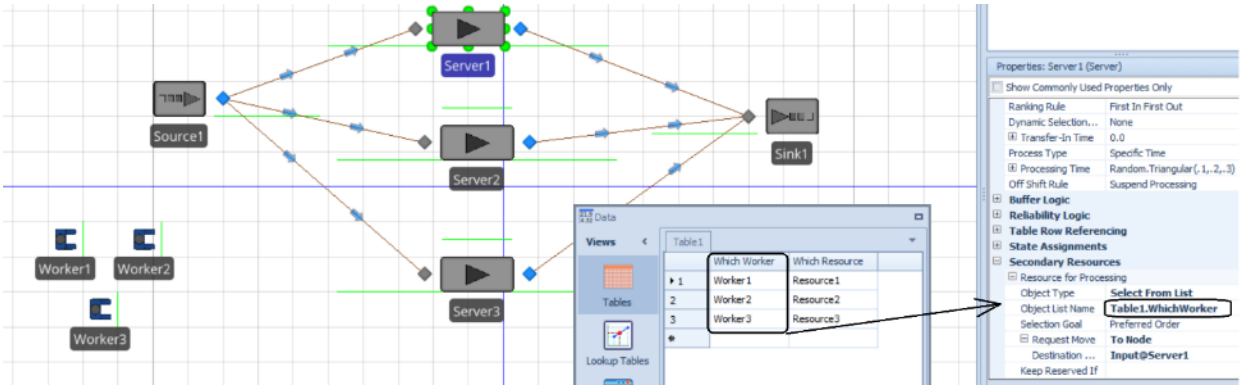
Simio Release 9 – Sprint 156 – June 2, 2017

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

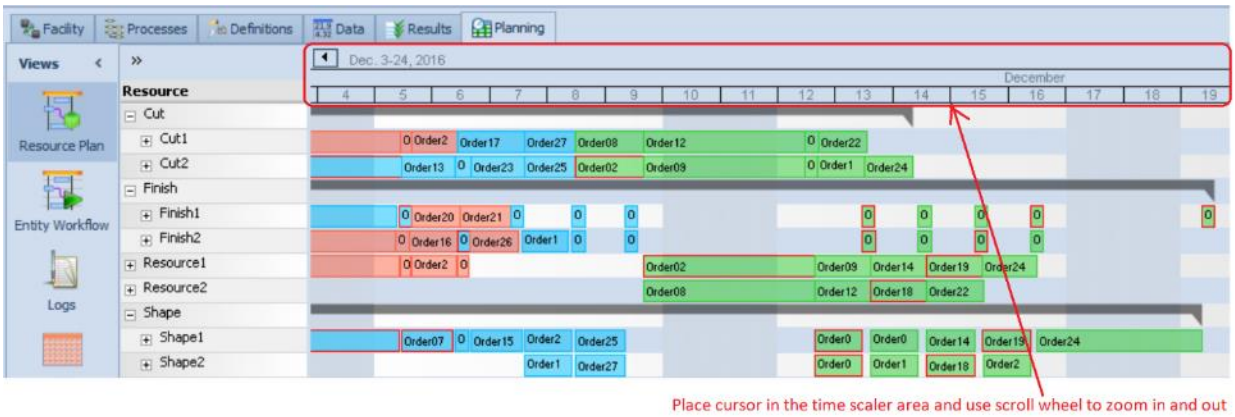
We have added the capability within several steps to allow for list members to also be specified within tables (in addition to Lists). The Seize, Move and Release steps now all allow for the TableName.ColumnName reference to be used as the *Object List Name* property value. Note that this table reference 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.

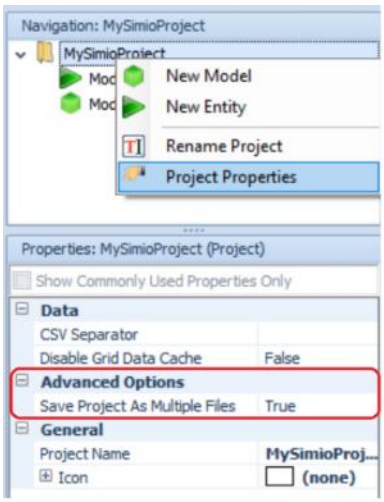


New SimBit
We have added a new reneging type SimBit to our Sample SimBit Solutions.

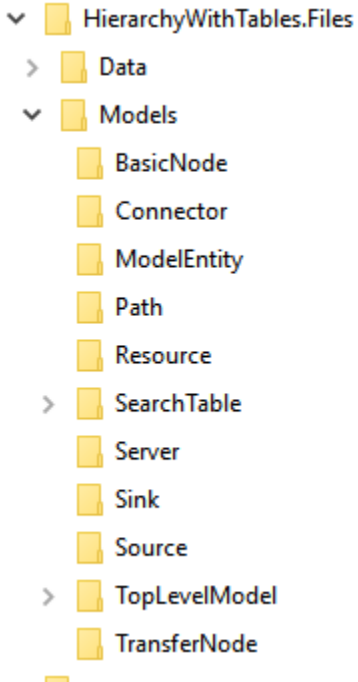
RenegingCombiner – this model illustrates a palletizer or similar machine (Combiner object) that batches a standard number of parts together. After the parent (pallet) entity has waited for a given amount of time for the individual parts to arrive, it will renege, reset the batch size to the number of 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 spfx), 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 has 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.

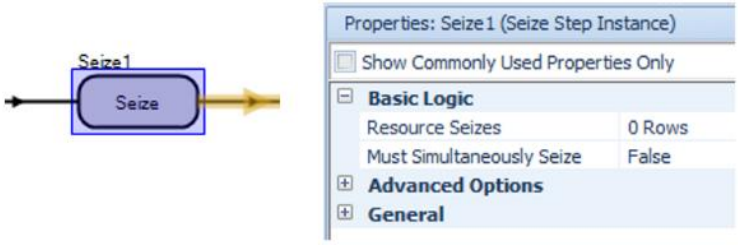
Academic Licensing

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.

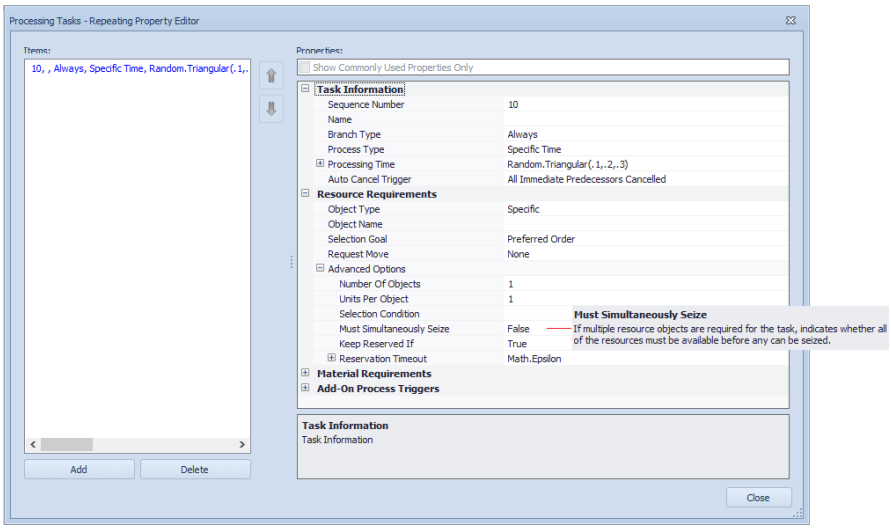


Seize Step – Enhanced Trace
In addition to the above Seize step changes, we have also enhanced the Simio 'trace' for the step so that users may more easily view why a resource or group of resources is seized or not seized. In the below screenshot, for example, the entity requires both Tool3 and Worker1. 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.

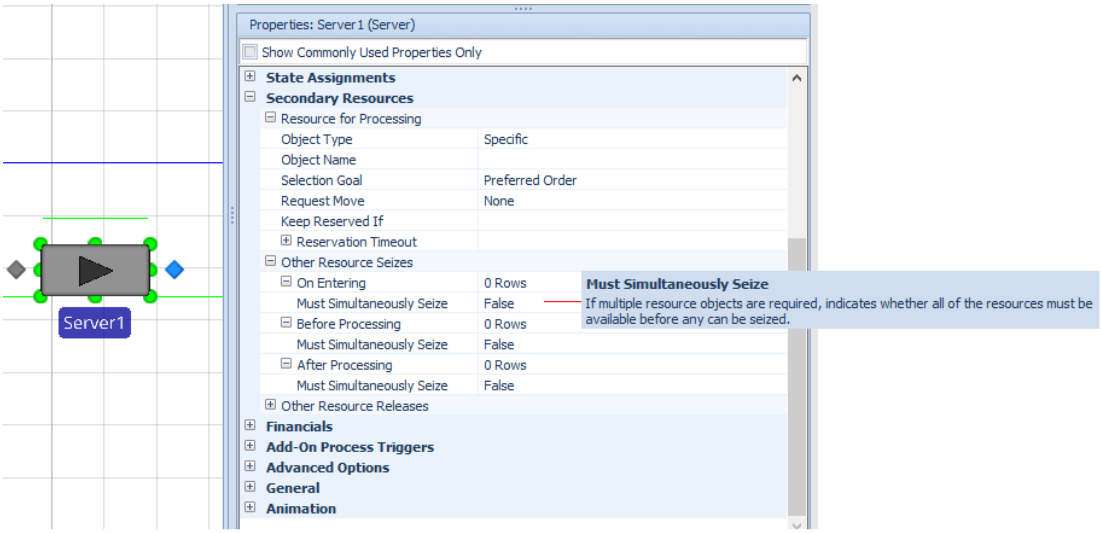
[Seize] Seize2	Stopped at breakpoint on Process1.Seize2
	Object 'DefaultEntity.34' checking availability to seize '1' resource(s) of resource type '[Specific] Tool3'.
	Object 'DefaultEntity.34' found '1' currently available resource(s) of resource type '[Specific] Tool3'.
	Object 'DefaultEntity.34' checking availability to seize '1' resource(s) of resource type '[Specific] Worker1'.
	Object 'DefaultEntity.34' unable to seize '1' capacity unit(s) of resource 'Worker1[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*.

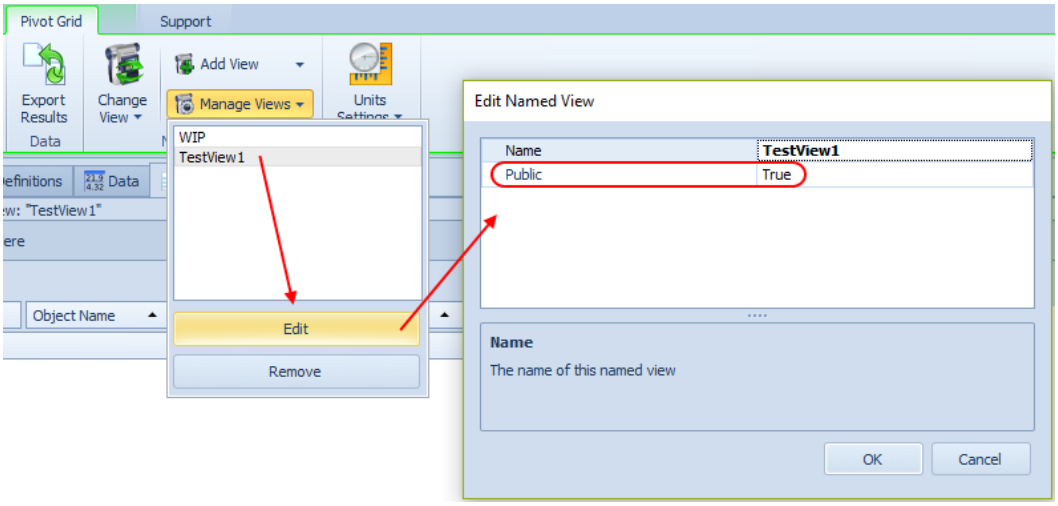


Simio Release 9 – Sprint 154 – April 10, 2017

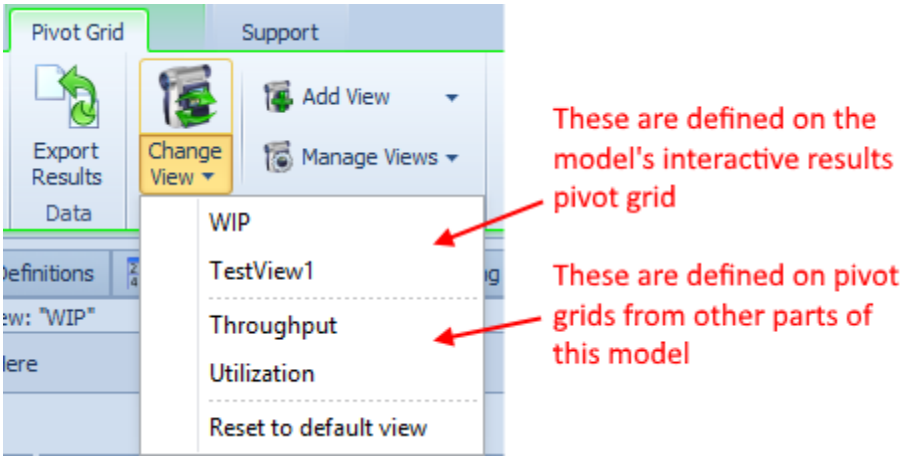
In this sprint, we have added the much-requested feature of *public* views for the pivot grids in different areas of the product. This allows users to customize pivot grid reports in a particular window (i.e., interactive results window), specify them as 'Public' and then view those same customized views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

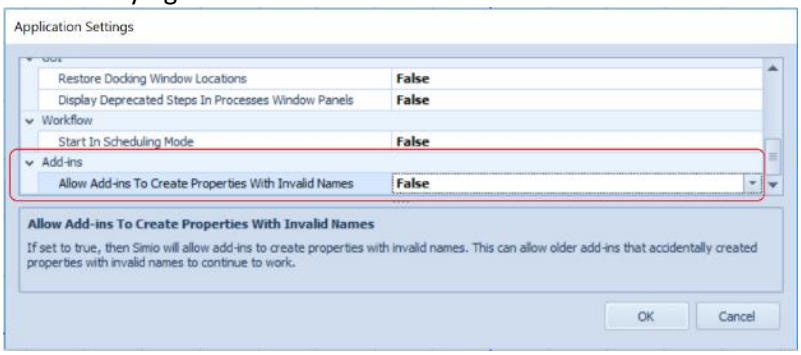


When applying a named view to a pivot grid, we will automatically hide or show certain columns as appropriate to the pivot grid. Specifically, the Minimum, Maximum, Half Width, and Std. Dev. columns will be hidden when applying an experiment's named pivot view to a model's pivot grid, while the

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.

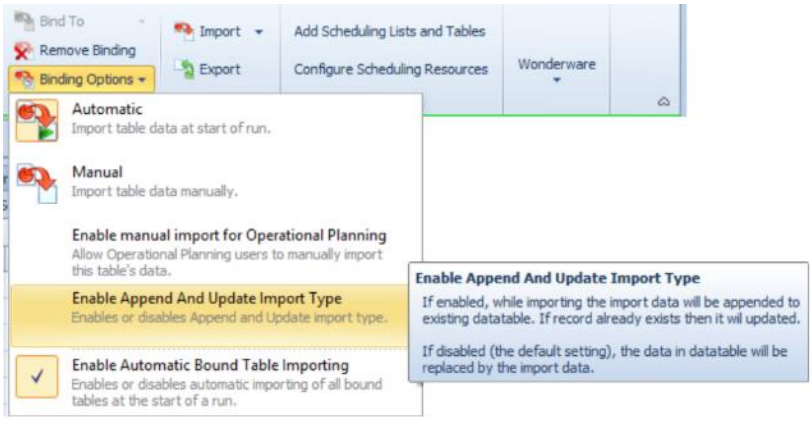
New Schedule Functions
The below schedule based functions have been added to allow users more flexibility accessing schedule based information. These functions can be accessed using ScheduleName.*, SchedulePropertyName.Schedule.* (e.g., ResourceObjectName.WorkSchedule.Schedule.* if the resource is following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

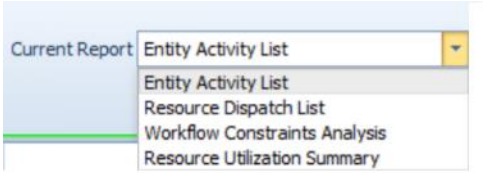


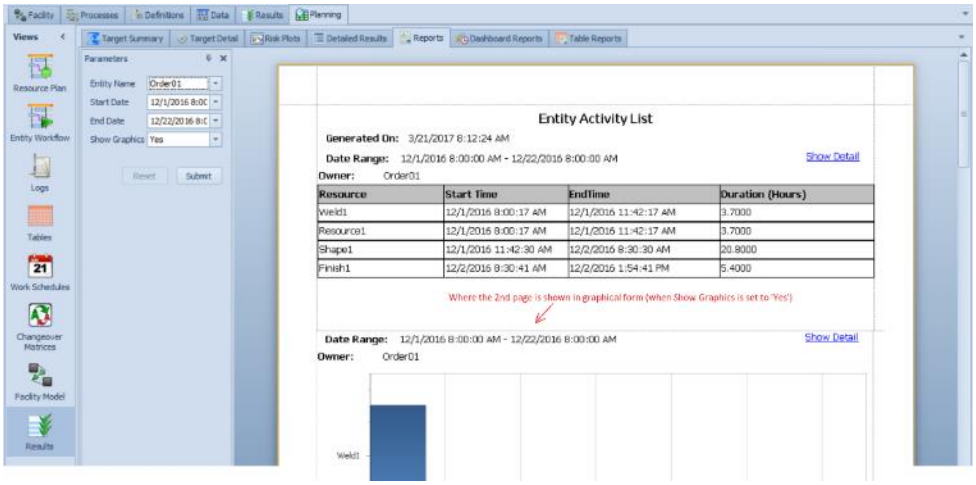
Tables - Sequence Destination Property
A standard Data Table can now be changed into a Sequence Table by using the new 'Sequence Destination' property type. This can be done by adding a new column of that type or changing an

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.



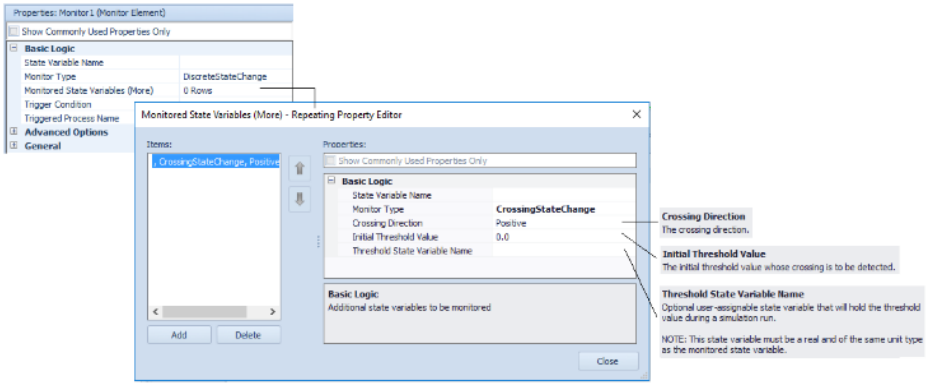


Simio Release 9 – Sprint 152 – February 24, 2017

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.

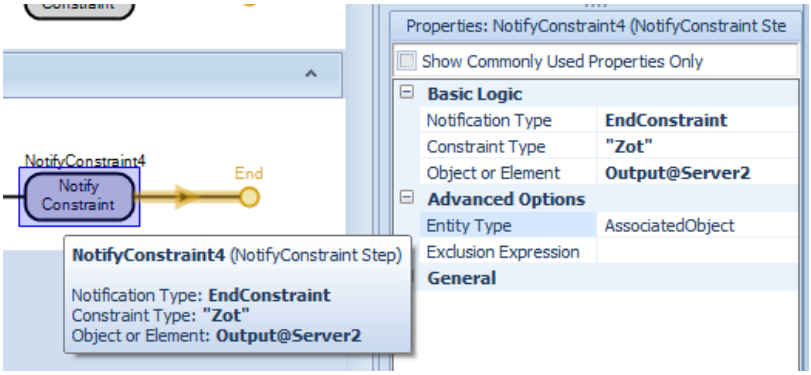
Monitor Element Enhancement
We have enhanced the Monitor element to be able to monitor more than one state variable, which will simplify the event-driven waiting approach (Wait step). The enhancement will make it easier to model a single 'Status Changed' event that pertains to some logical grouping of 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.spfx) 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):

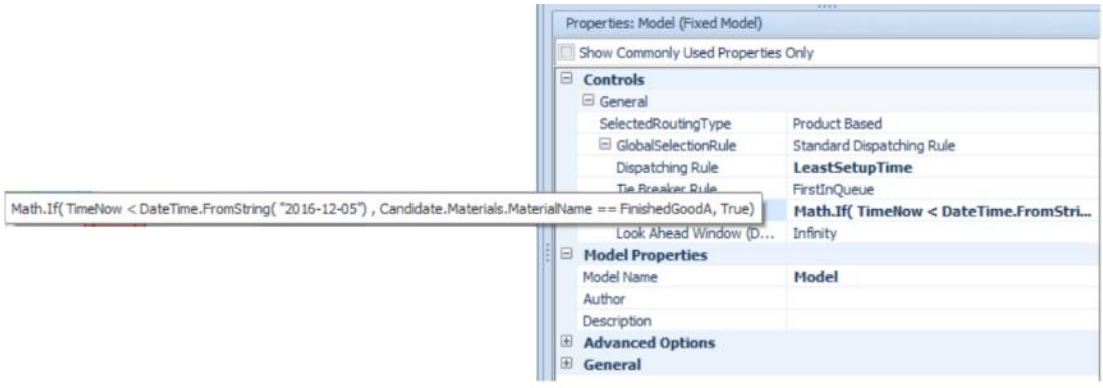
Resource Usage Log | Resource State Log | Resource Capacity Log | Constraint Log | Transporter Usage Log | Material Usage Log | Task Log | State Observation Log

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

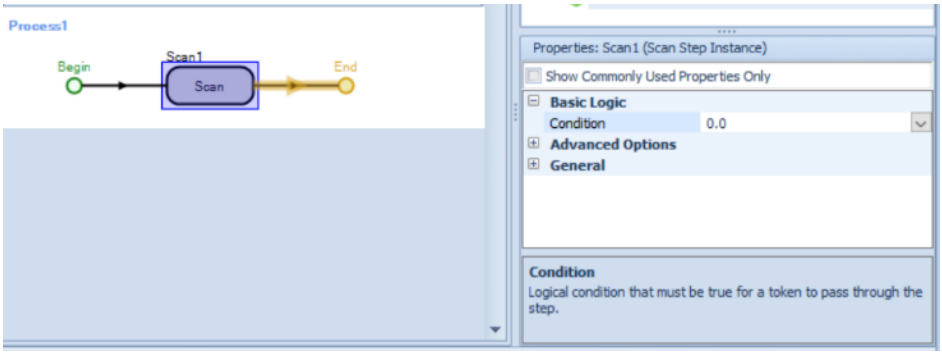
The Dynamic Selection Rule grouping of properties has been enhanced to include a *Filter Expression*. This allows a user to optionally filter out any candidate entities that don't satisfy a specified logical condition. It is like the *Filter Expression* property provided by the 'Smallest Value First'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

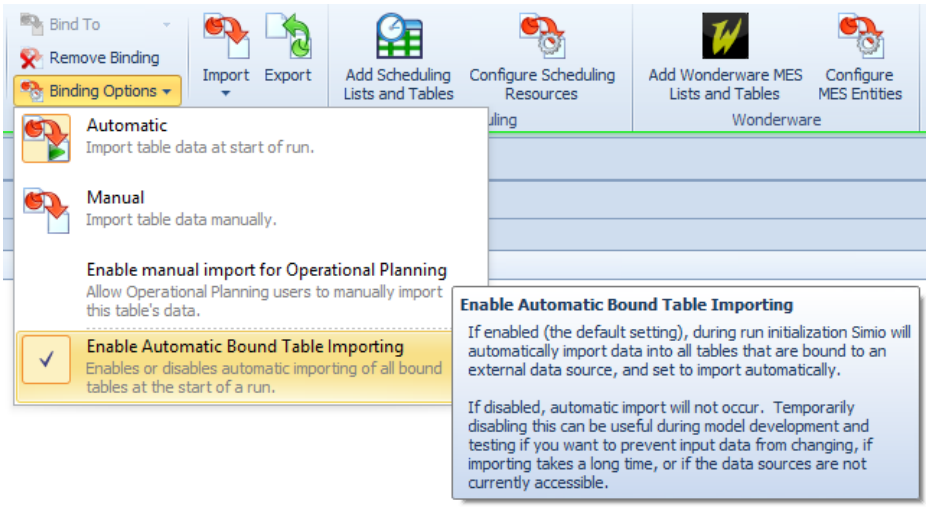
New Scan Step
We have added a Scan step which provides a new polled waiting approach that allows a modeler to hold a process token until a specified condition is true, where the logical condition may be specified as any arbitrary expression.



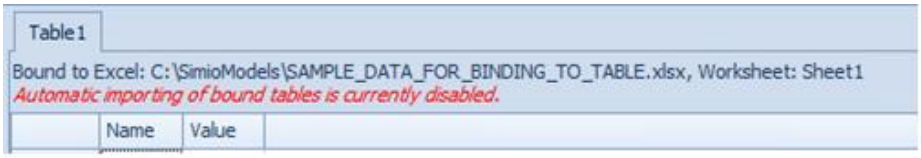
See more details in the Simio help for specifics of how this step works.

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.



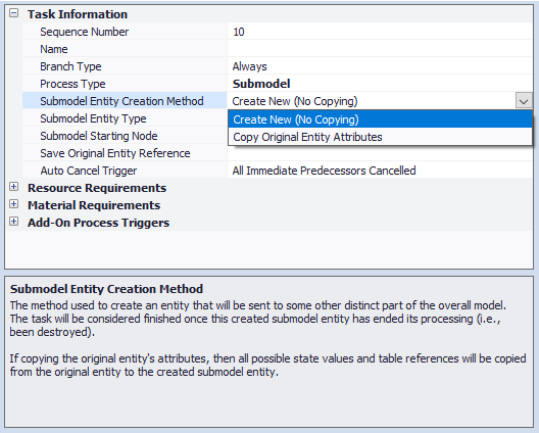
Note that this does not "temporarily *remove* table bindings" but will bypass the automatic import at start of run. When a model is in this state, bound tables that are set to automatically import show the following in red:



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.



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.

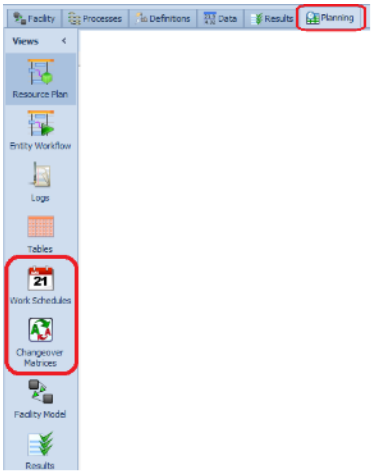
String.CompareSequenceNumbers(sequenceNumber1, sequenceNumber2) - Compares two specified sequence number strings and returns 0 if the two sequence numbers have no implied dependency relationship, -1 if an item assigned the first sequence number must come before an item assigned the second sequence number,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



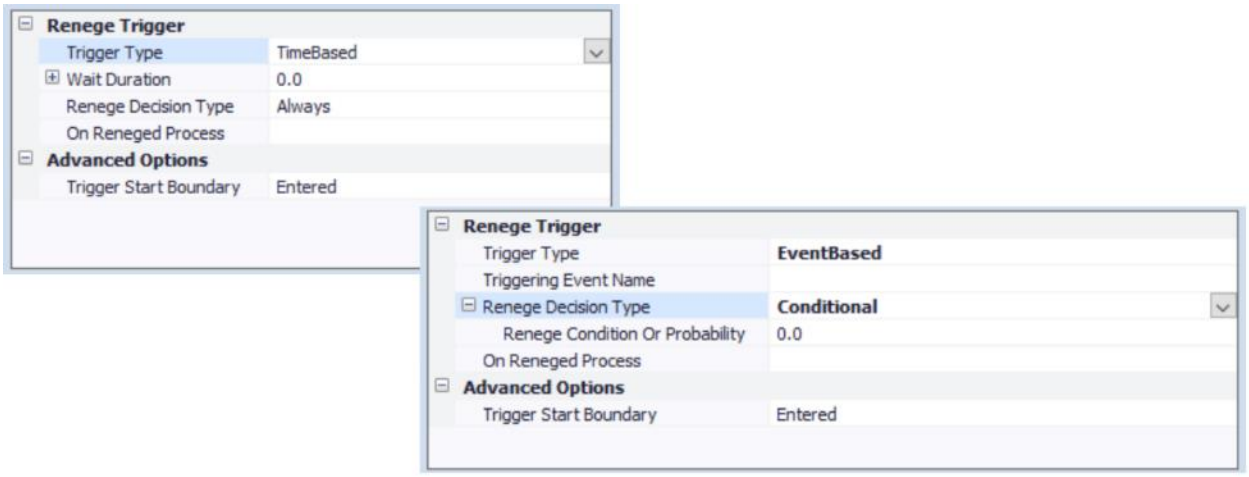
Updated Visual Studio Templates
We have updated our C# Visual Studio Templates to support Visual Studio 2012, 2013 and 2015, and have added corresponding templates for VB.NET.

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 reneging, or abandoning waiting in a queue or station. Models illustrating various types of balking and reneging have also been added to our extensive SimBit library.

Station Element - Reneging

The Station element has been enhanced to include reneging, 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 reneging. Triggers can be Time or Event based, and once the trigger has occurred, the reneging can happen always or based on a probability or condition of the system.

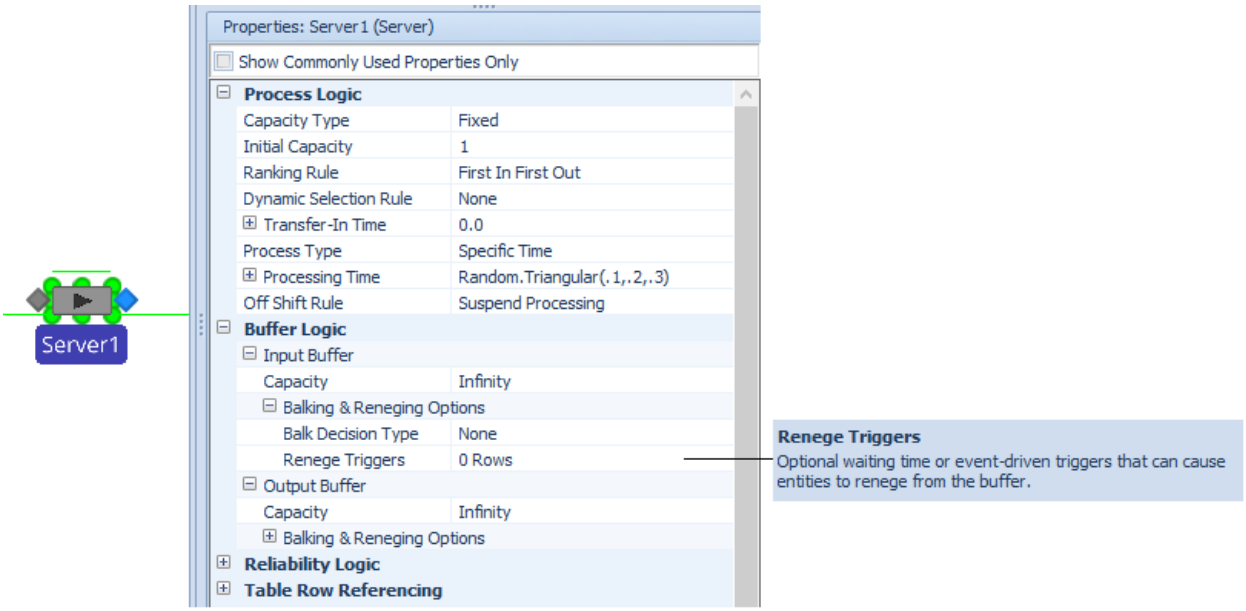


There is also a new function, **NumberReneged**, that returns the total number of entities that have abandoned waiting in the station.

Buffer Logic – Reneging 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 reneging options for the buffer(s) within the Source, Server, Combiner, Separator and Workstation within the Standard Library, as well as the Filler, Emptier, ItemToFlowConverter and FlowToItemConverter within the Flow Library.

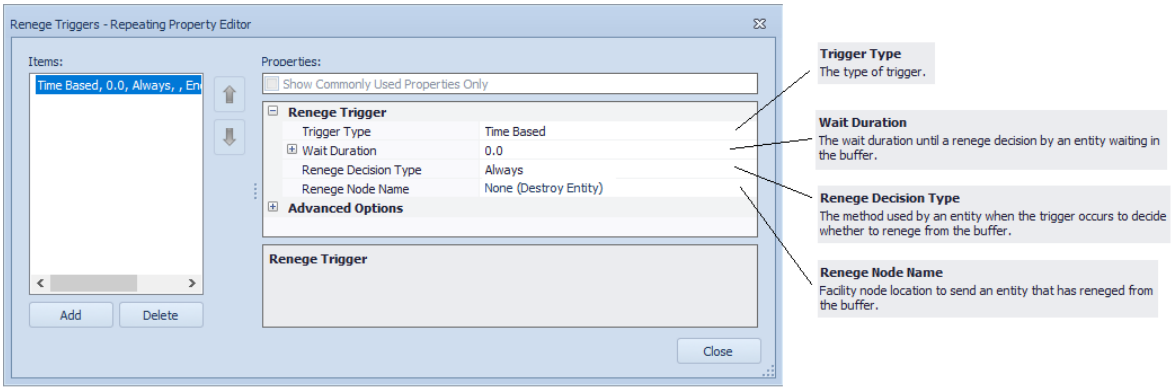
For each input buffer and output buffer, the Balking & Reneging Options includes the balking properties as well as a Renege Triggers repeating property editor to specify one or more methods to trigger reneging within a buffer.

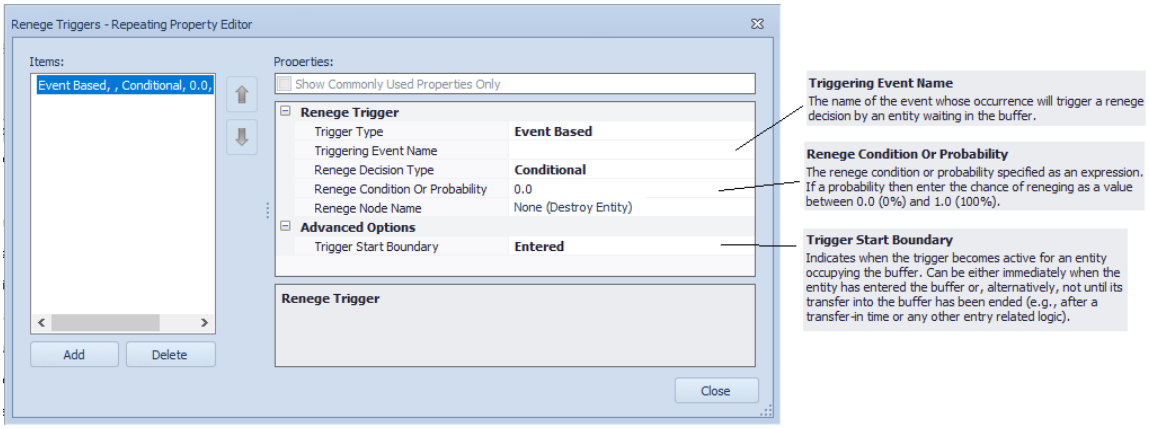


Within the Renege Triggers repeating property editor, there is a *Trigger Type* property to allow for either 'Time Based' or 'Event Based' triggering. For Time Based triggering, a *Wait Duration* expression is specified, while for Event Based triggering, the *Triggering Event Name* property is displayed.

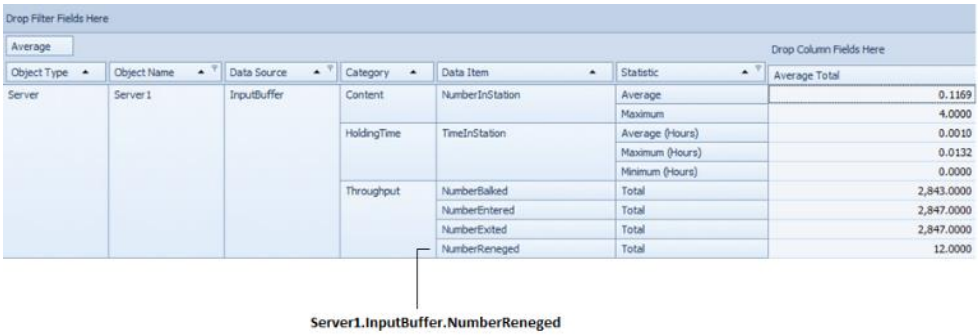
If reneging is triggered, the action can happen either always (no conditions), or based on a probability or condition of the system. This allows for logic such as an entity evaluating the buffer size after a wait duration to determine whether to remain in the queue. The reneged entity can either be destroyed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



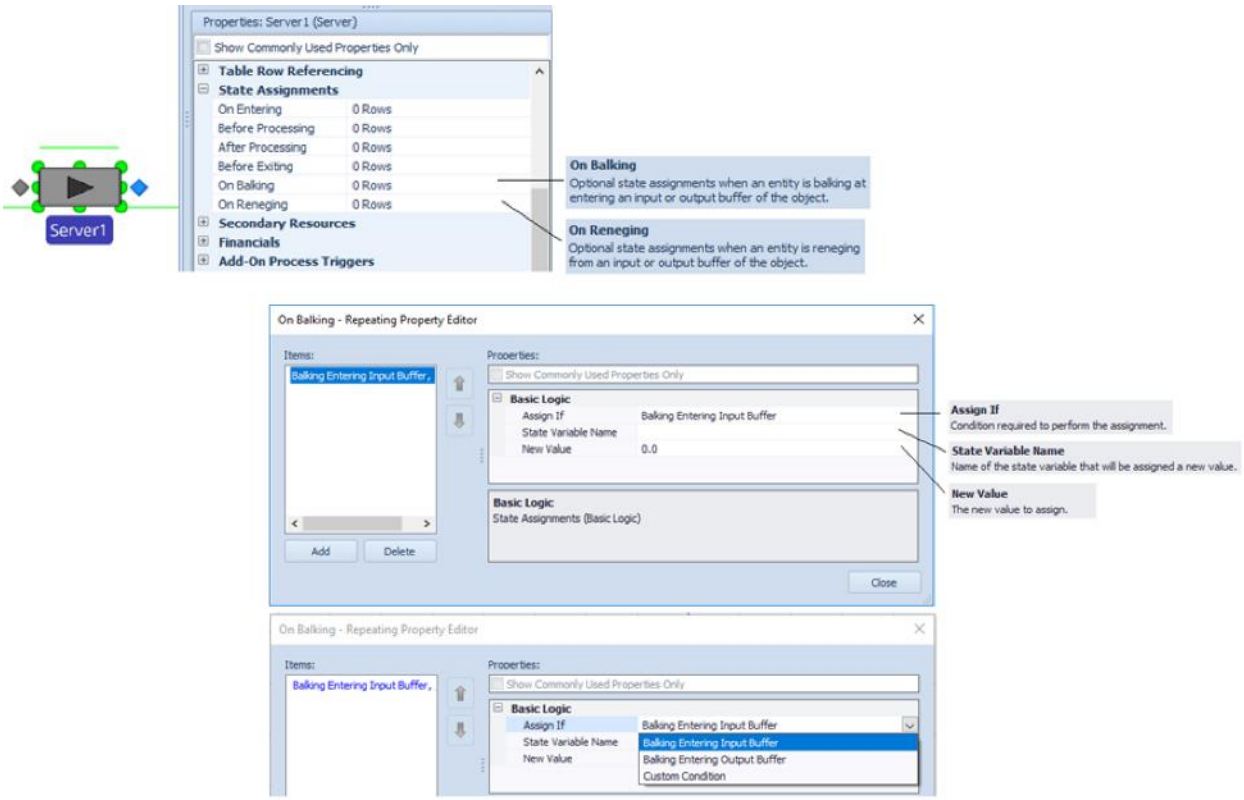


Simio also includes statistics on the total number of reneged entities for a buffer in the automatically reported results.



State Assignments - On Balking and On Reneging

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 reneging 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:

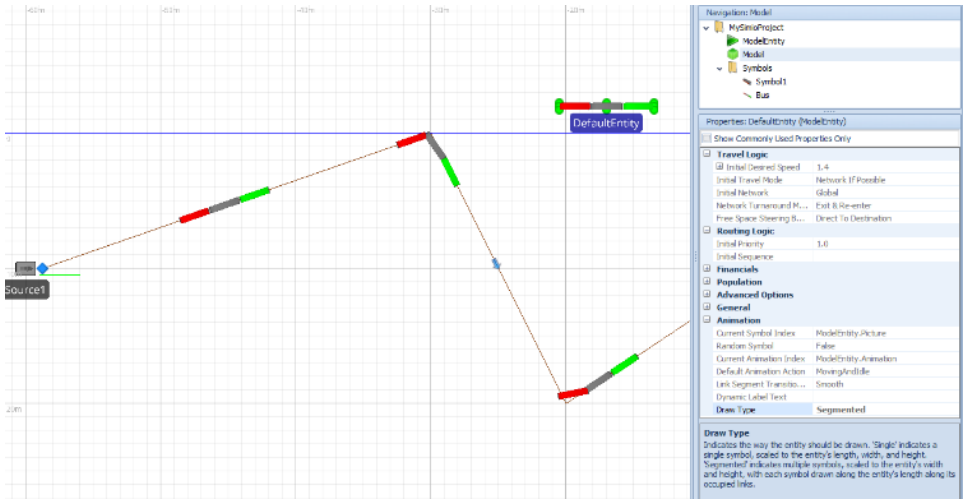
Queuing.IsWaitingResourceAllocationQueue - Returns True (1) if the entity is currently waiting in the AllocationQueue of a resource. Otherwise, the value False (0) is returned.
Queuing.IsWaitingMaterialAllocationQueue - Returns True (1) if the entity is currently waiting in the AllocationQueue of a material. Otherwise, the value False (0) is returned.
Queuing.IsWaitingStationEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a station. Otherwise, the value False (0) is returned.
Queuing.IsWaitingLinkEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a link. Otherwise, the value False (0) is returned.
Queuing.IsWaitingNodeEntryQueue - Returns True (1) if the entity is currently waiting in the EntryQueue of a node. Otherwise, the value False (0) is returned.
Queuing.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.
Queuing.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.
Queuing.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.
Queuing.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.

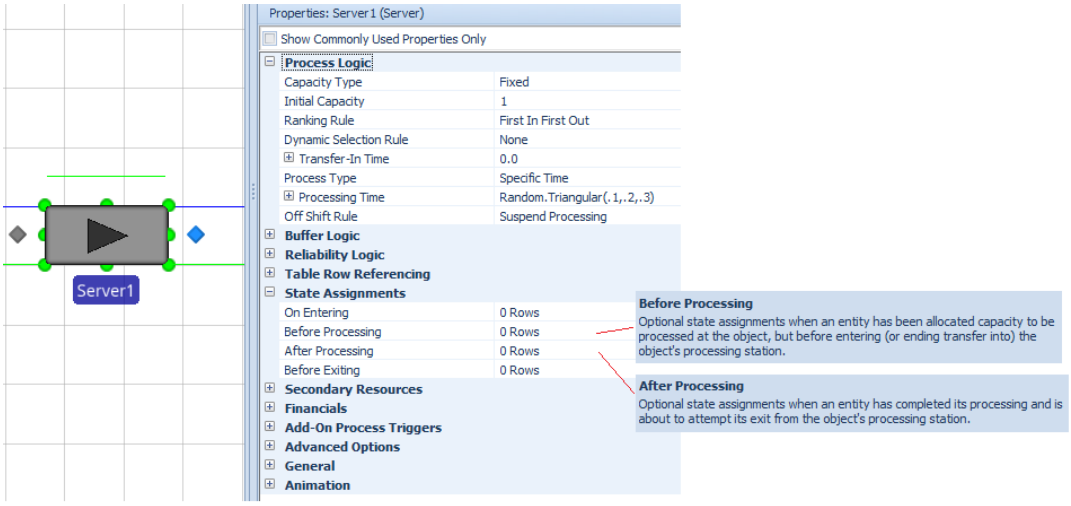
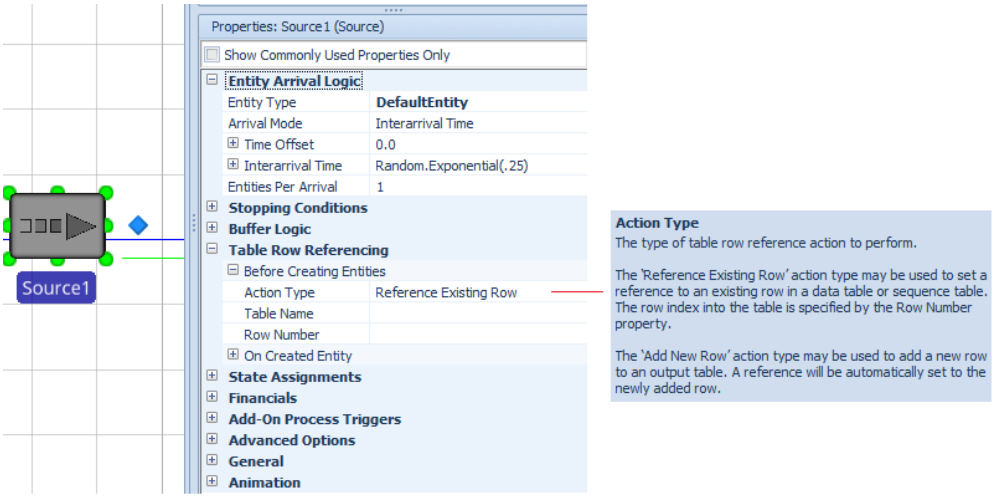
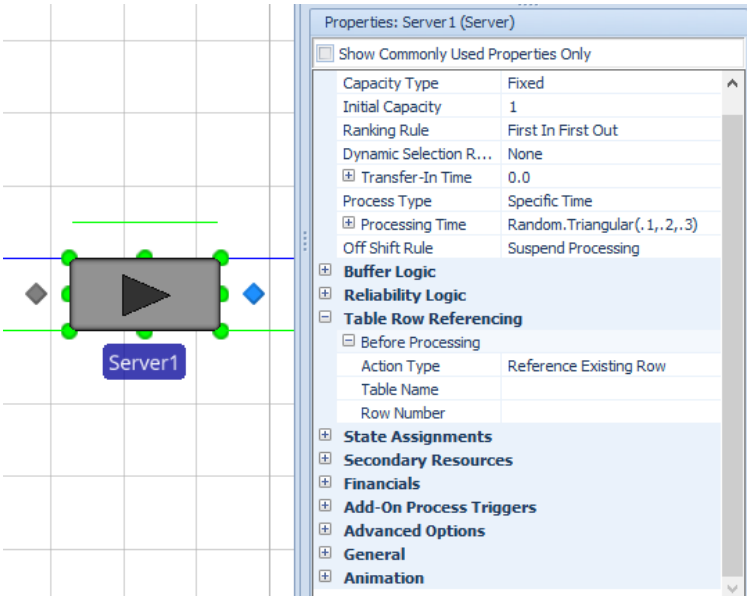


Table Row Referencing in Library Objects

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.



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.

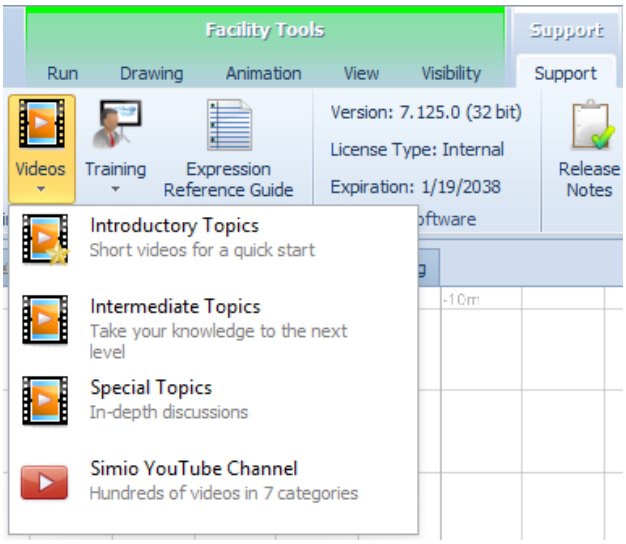


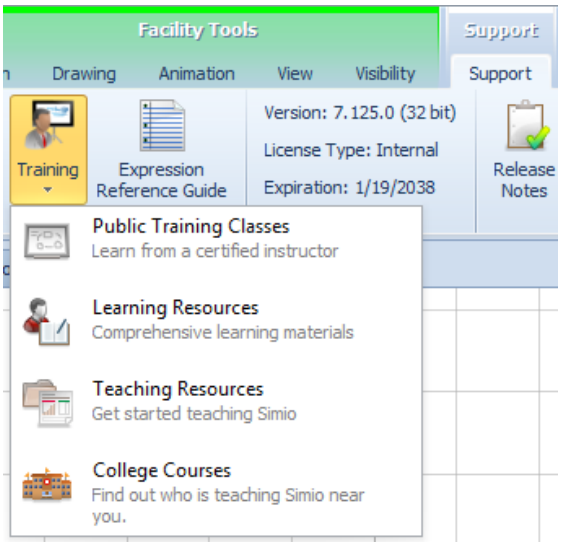
SetRow and AddRow Step Enhancements

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.



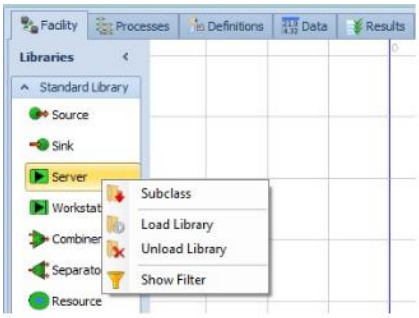


Remove Step Enhancements

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.

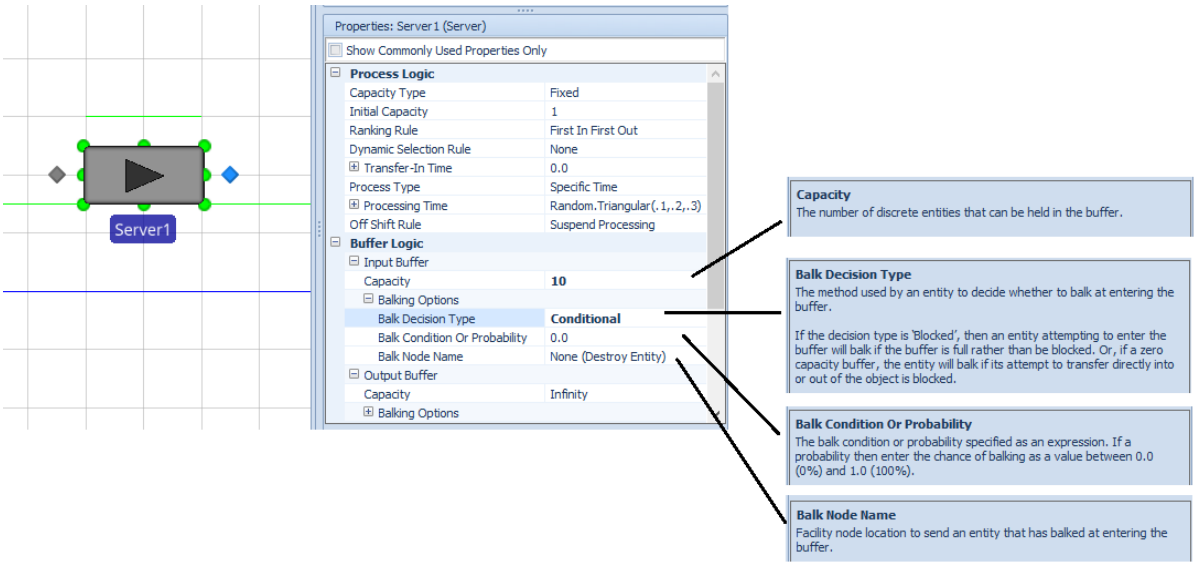
Simio Release 8 – Sprint 146 – October 12, 2016

In this sprint, we have enhanced our objects containing input and output buffers in both the Standard and Flow Libraries with balking options, with many options for conditional and probabilistic balking evaluation.

Buffer Logic – Balking Options

We have added balking options to the input and/or output buffers for the Source, Server, Combiner, Separator and Workstation within the Standard Library. Additionally, these features have been added to the Filler, Emptier, ItemToFlowConverter and FlowToItemConverter within the Flow Library.

For each input buffer and output buffer, the Balking Options includes the *Balk Decision Type*, which can be 'None', 'Blocked', 'Conditional' or 'Probabilistic', as well as the *Balk Condition or Probability* and the *Balk Node Name* which can be a specific node or 'None (Destroy Entity)'. A new Balking Options page in the Simio help provides examples for the different property options.



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

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
Server	Server1	InputBuffer	Content	NumberInStation	Average	1.4290
					Maximum	15.0000
			HoldingTime	TimeInStation	Average (Hours)	0.0060
					Maximum (Hours)	0.0472
					Minimum (Hours)	0.0000
			Throughput	NumberBalked	Total	1,971.0000
				NumberEntered	Total	5,688.0000
				NumberExited	Total	5,685.0000
		OutputBuffer	Throughput	NumberEntered	Total	5,684.0000
				NumberExited	Total	5,684.0000
		Processing	Content	NumberInStation	Average	0.7874
					Maximum	1.0000
			HoldingTime	TimeInStation	Average (Hours)	0.0033
					Maximum (Hours)	0.0050
					Minimum (Hours)	0.0017
			Throughput	NumberEntered	Total	5,685.0000
				NumberExited	Total	5,684.0000

Server1.InputBuffer.NumberBalked

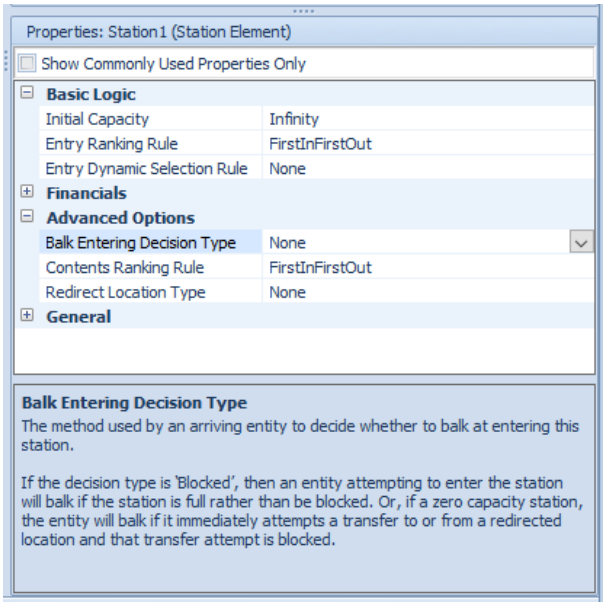
Random Number Generator

We fixed a minor problem with the low-level implementation of the RNG. This may cause slightly different results to be returned. See Compatibility Notes for more details.

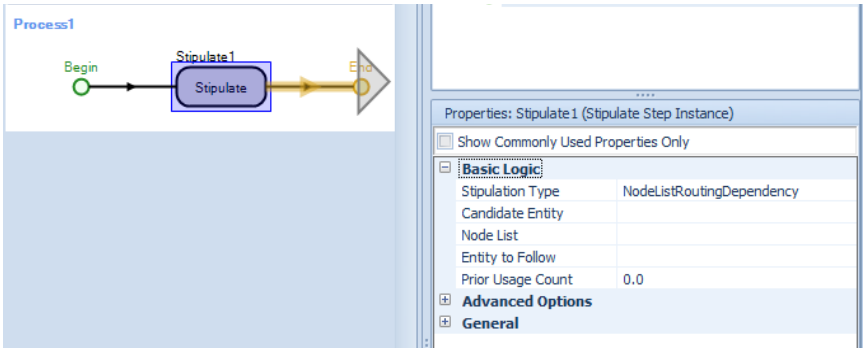
Simio Release 8 – Sprint 145 – September 23, 2016

In this sprint, we have started progress on allowing balking within buffers by enhancing the Station element, added several new SimBits and made additional user-requested enhancements.

Station Enhancement – Buffers with Balking
We have added some new properties to the Advanced Options for the Station element, including the *Balk Entering Decision Type* and *Redirect Location Type* with supporting properties. These will support modelling of buffers with balking using Standard Library objects, which is a much requested feature we are adding currently over the next 1-2 sprints.



New Stipulate Step -
The Stipulate step may be used to add dependencies and requirements that entities must satisfy before proceeding. Stipulations can be for node list selection or resource allocation. It is primarily intended for use with Simio Enterprise Edition and has similar functionality (except through Processes window) to the Gantt drag and drop feature.



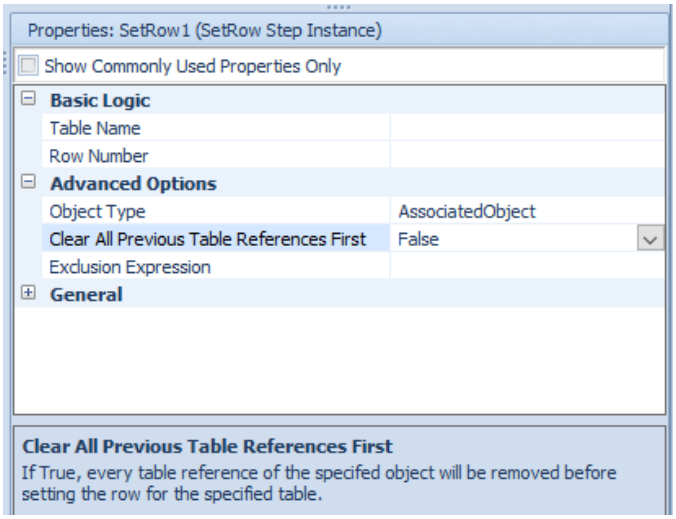
New SimBits – Stipulate Step

We've added 2 new SimBits to demonstrate the use of the Stipulate step. These projects include models showing standard use of resources and node lists with no stipulations, as well as drag and drop Gantt stipulations and use of the Stipulate steps to emulate the drag and drop functionality.

UsingTheStipulateStepForNodeLists.spfx – This project contains three models to demonstrate the various uses of node lists, including the drag and drop in the Resource Plan Gantt for node list routing dependencies and node list requirements, as well as the use of the Stipulate step to determine those list requirements.
UsingTheStipulateStepForResourceAllocation.spfx – This project contains three models to demonstrate the various types of resource allocation, including the drag and drop in the Resource Plan Gantt for resource allocation dependencies, as well as the use of the Stipulate step to determine resource allocation requirements.

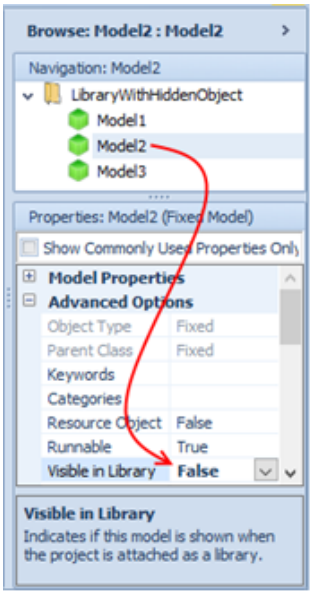
Performance Improvements
Users may see some performance improvements under certain circumstances when using repeating graphic symbols in the Facility window.

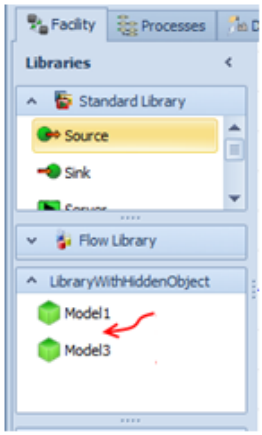
SetRow Step Enhancement – Clear Table References
There is a new property on the SetRow step that allows users to clear all previous table references for an object prior to setting a row for that object for a given table. This has been a much requested feature from users.



New Entity Function
Entity.IsTravelingInReverse - Returns True (1) if the entity is currently traveling in reverse. Otherwise, the value False (0) is returned. This function returns True if an entity is traveling in forward orientation on a network and then does a turnaround on a bidirectional link using the *Network Turnaround Method* 'Reverse'. It will also return True if an entity is traveling in free space with a negative movement rate.

Visibility Flag for Models in Projects
We have added a new property, *Visible in Library*, which allows users to 'hide' a particular model within a project when that project is loaded as a library. This was requested by several users building custom objects.


Browse: Model2 : Model2
Navigation: Model2
LibraryWithHiddenObject
Model1
Model2
Model3
Properties: Model2 (Fixed Model)
Show Commonly Used Properties Only
Model Properties
Advanced Options
Object Type
Fixed
Parent Class
Fixed
Keywords
Categories
Resource Object
False
Runnable
True
Visible in Library
False
Visible in Library
Indicates if this model is shown when the project is attached as a library.


Facility
Processes
Libraries
Standard Library
Source
Sink
Flow Library
LibraryWithHiddenObject
Model1
Model3

Simio Release 8 – Sprint 144 – August 29, 2016

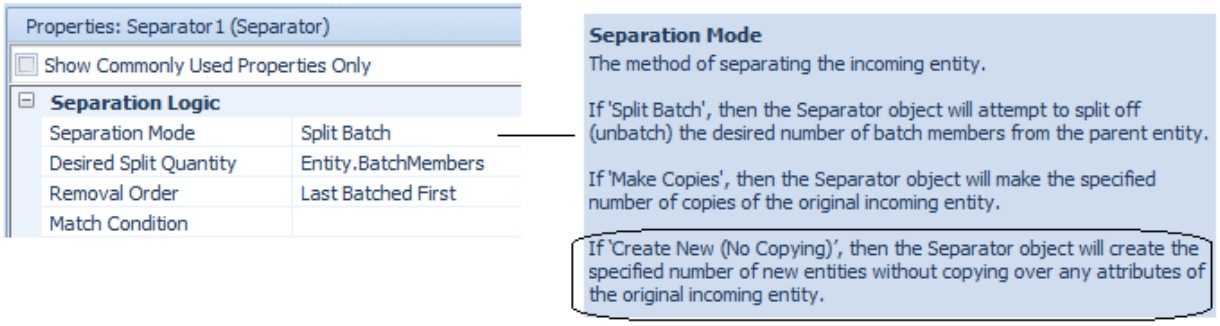
In this sprint, we have enhanced some of our current objects with new features, including adding new 'setup' related states and statistics based on our setup enhancements from sprint 143, adding an option on Separator to make new entities, and allowing multiple stop early triggers on Filler and Emptier (Flow Library).

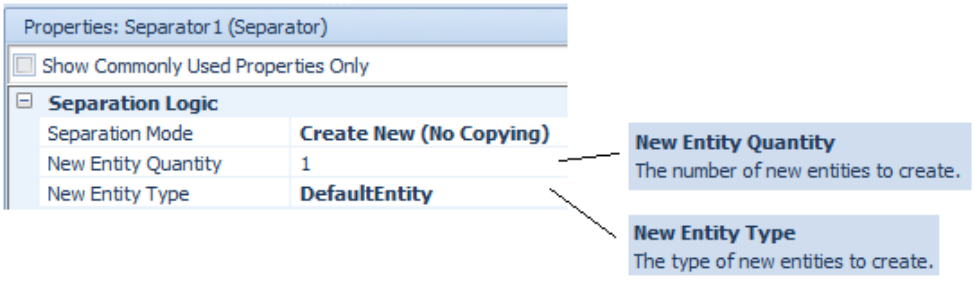
Server / Combiner / Separator –Setup and OffShiftSetup States
Last sprint 143, we added the option of sequence dependent setup task occurring at a Server, Combiner, or Separator. We've now enhanced the ResourceState state variable of those objects to be set to the value **Setup (7)** or **OffShiftSetup (8)**. The latter value if the object is offshift but the *OffShift Rule* is 'Finish Work Already Started' and thus the setup task is being finished while offshift.

Users will also now be able to optionally animate the Server, Combiner, or Separator object's symbol differently if it is in one of these new resource states, and ResourceState->TimeSetup and ResourceState->TimeOffShiftSetup statistics will be automatically reported for the object in the results. Additionally, these new resource states may be seen within the Resource Gantt (Simio Enterprise Edition). There was also one new function added for an object used as a resource:

HasActiveChangeover - Returns True (1) if there is an active changeover occurrence at the resource. Otherwise, the value False (0) is returned.

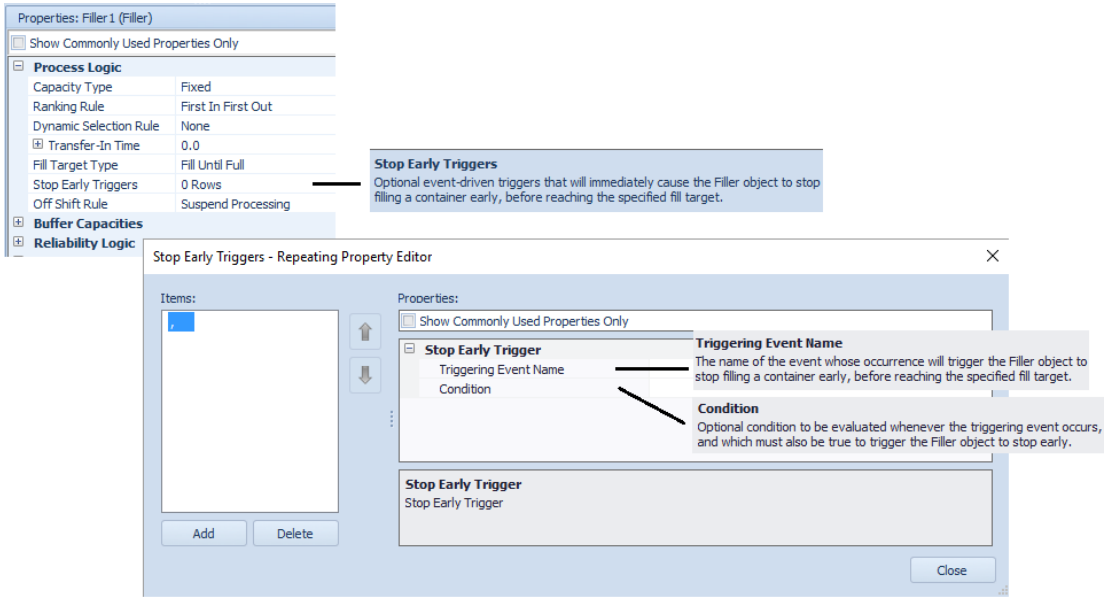
Separator Enhancement – Create New (No Copying)
We have added an additional *Separation Mode* option for the Separator object, allowing a user to simply create new entities upon processing each parent entity without copying over any state or attribute data.





Filler and Emptier Enhancement – Stop Early Triggers

In Sprint 137, we enhanced the FlowToItemConverter and ItemToFlowConverter objects to include a *Stop Early Triggers* repeat group using the simple Event-Condition-Action oriented design. We have now expanded this feature to the Filler and Emptier objects, as shown below. Note that the previous *Stop Early Event Name* property that both those objects used to provide has now been deprecated and moved into Advanced Options -> Deprecated Properties.



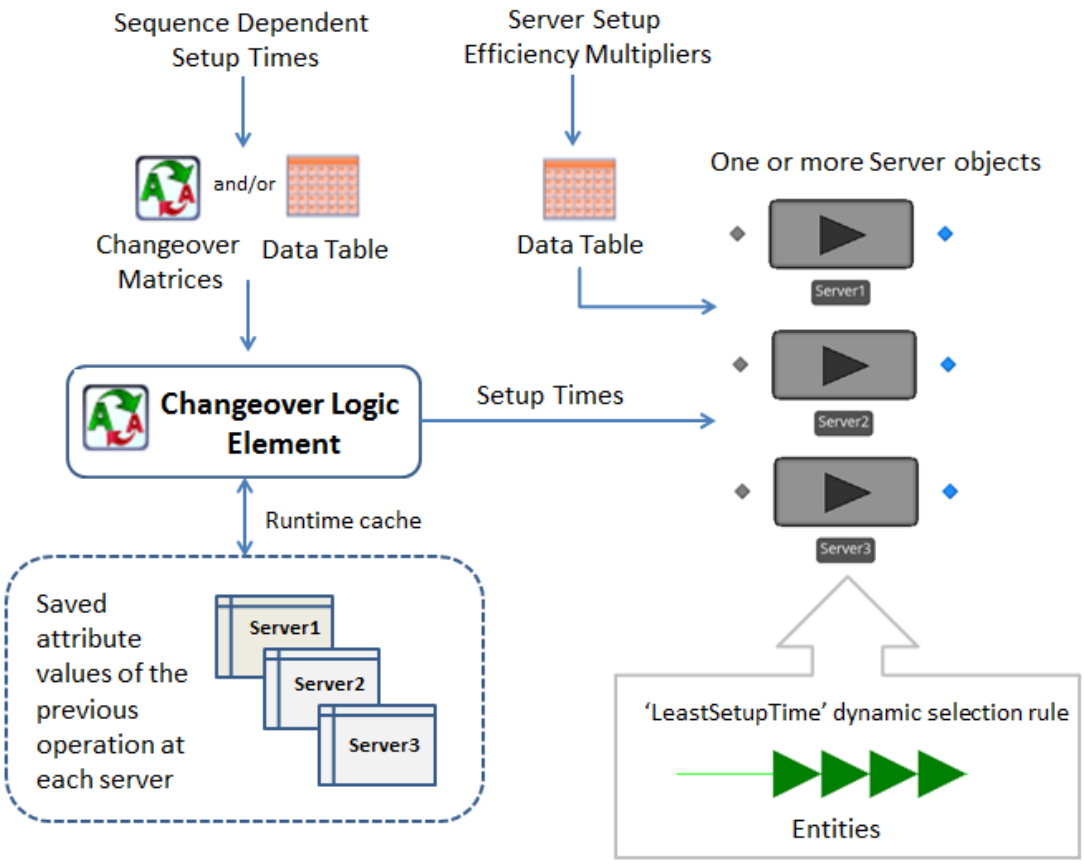
Simio Release 8 – Sprint 143 – August 10, 2016

In this sprint, we are very excited to share the latest feature within the Server type objects, namely Sequence Dependent Setups. Users can now easily specify setups and reference changeover matrices or tables within the Processing Tasks of the Task Sequences section for Server, Combiner and Separator objects. Additionally, we have continued to enhance a few features, such as work schedules (new step to dynamically change) and data tables (can 'split' for easy editing).

Modeling Servers with Sequence Dependent Setups – General Information
We have made many enhancements to support the use of sequence dependent setups within Server / Combiner / Separator objects. The below diagram illustrates a typical modelling scenario where there are sequence dependent setups at one or more servers in the system. The specific enhancements are then described in more detail in the separate sections below.

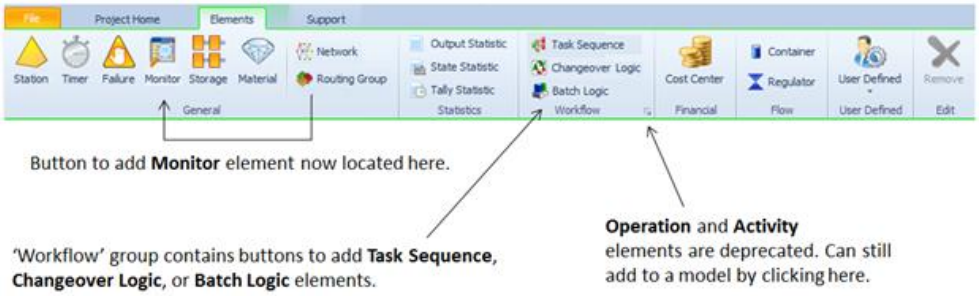
Sequence dependent setup time data is defined in changeover matrices and/or data tables. That setup time data is then referenced by a **Changeover Logic** element which is being used by either a single server or a group of servers. If there are multiple servers using the same changeover logic, then those individual servers may have different setup efficiency multipliers where each setup is possibly performed slower or faster at a particular server than the average time.

During the model run, entities representing jobs might be selected for processing at the servers using the 'LeastSetupTime' dispatching rule. The **Changeover Logic** element takes care of caching the attribute values of the previous operation at each server in order to determine and return the next sequence dependent setup time.

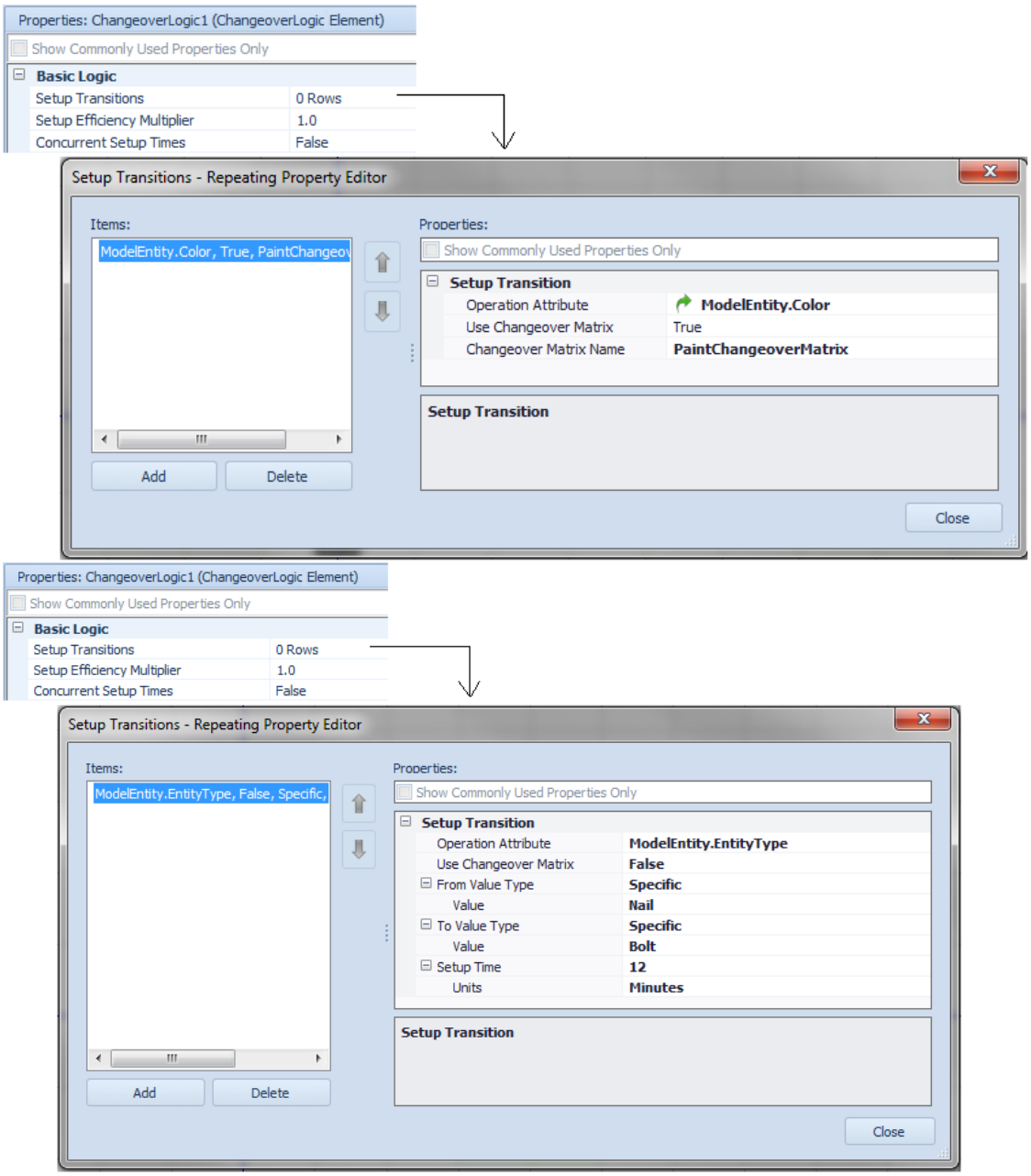


New ChangeoverLogic Element and Changeover Step

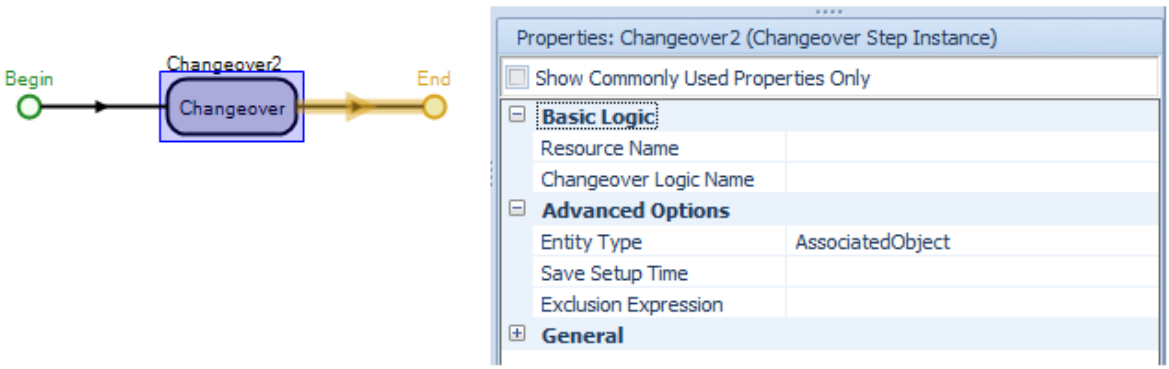
We have added a new ChangeoverLogic element that can be used in conjunction with the new Changeover step to model sequence dependent setups at one or more resources. Below shows the new Elements ribbon.



There are many ways that the Setup Transitions repeating properties of the Changeover Logic element can be defined, including referencing a Changeover Matrix (first diagram) as well as specific attribute to attribute values (second diagram). Many examples are shown in the discussions section of help for this element.

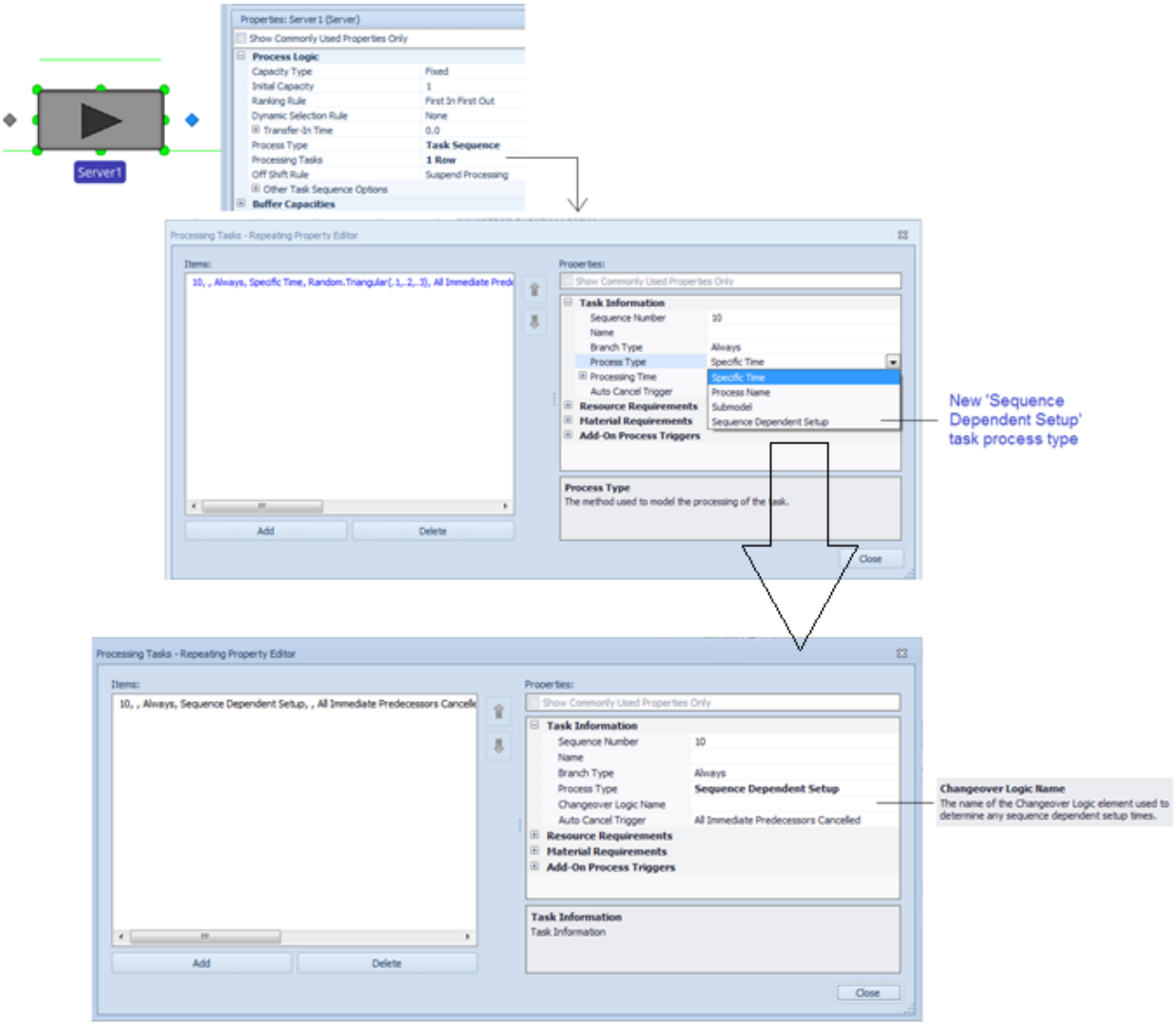


The new Changeover step can be used in conjunction with the ChangeoverLogic element to model sequence dependent setups for a particular resource.



Enhanced Server / Combiner / Separator - Task Sequence Process Type

Now included with Servers, Combiners and Separators is the option for 'Sequence Dependent Setup' under the *Process Type* property of the Processing Tasks repeating properties for Task Sequences.



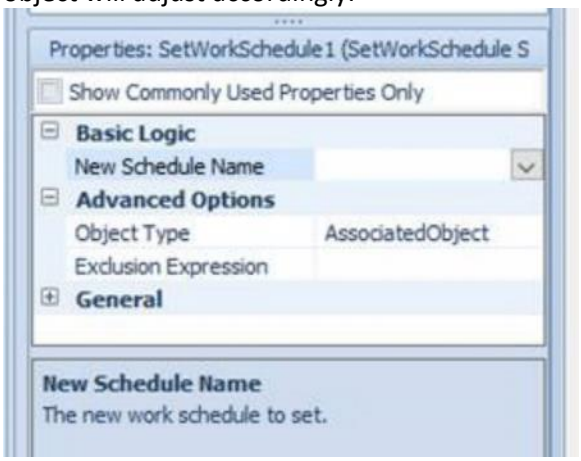
New SimBits – Sequence Dependent Setups

ServerUsingTaskSequenceWithSequenceDependentSetups.spfx – This model is a modification of the WorkstationWithSequenceDependentSetup SimBit. It demonstrates the use of the Server's sequence dependent setup with multiple entity (attribute - color) types, a ChangeoverLogic element and a Changeover matrix.

ServersUsingTaskSequenceWithDataTables_SequenceDependentSetups.spfx – This model also demonstrates the use of the sequence dependent setup options within a Server. This SimBit includes processing task data and setup transitions specified within relational data tables.

New SetWorkSchedule Step and Associated Function

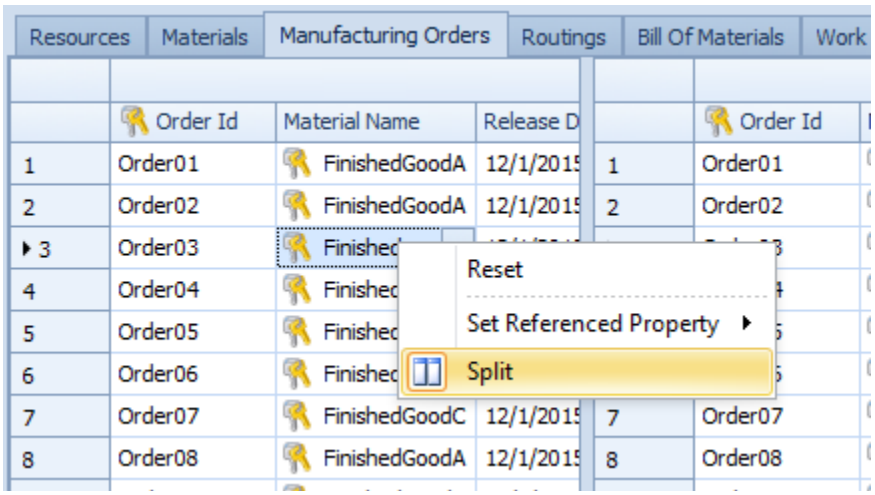
Following up the work schedule enhancements from the previous sprint (addition of Table Based), we have added a new step that will allow the user to dynamically assign the active work schedule for an object at runtime. Based on the current simulation and the newly assigned schedule, the capacity of the object will adjust accordingly.



The following function has also been added, **CurrentWorkSchedule.Name,** which will return the currently active schedule for a given resource object.

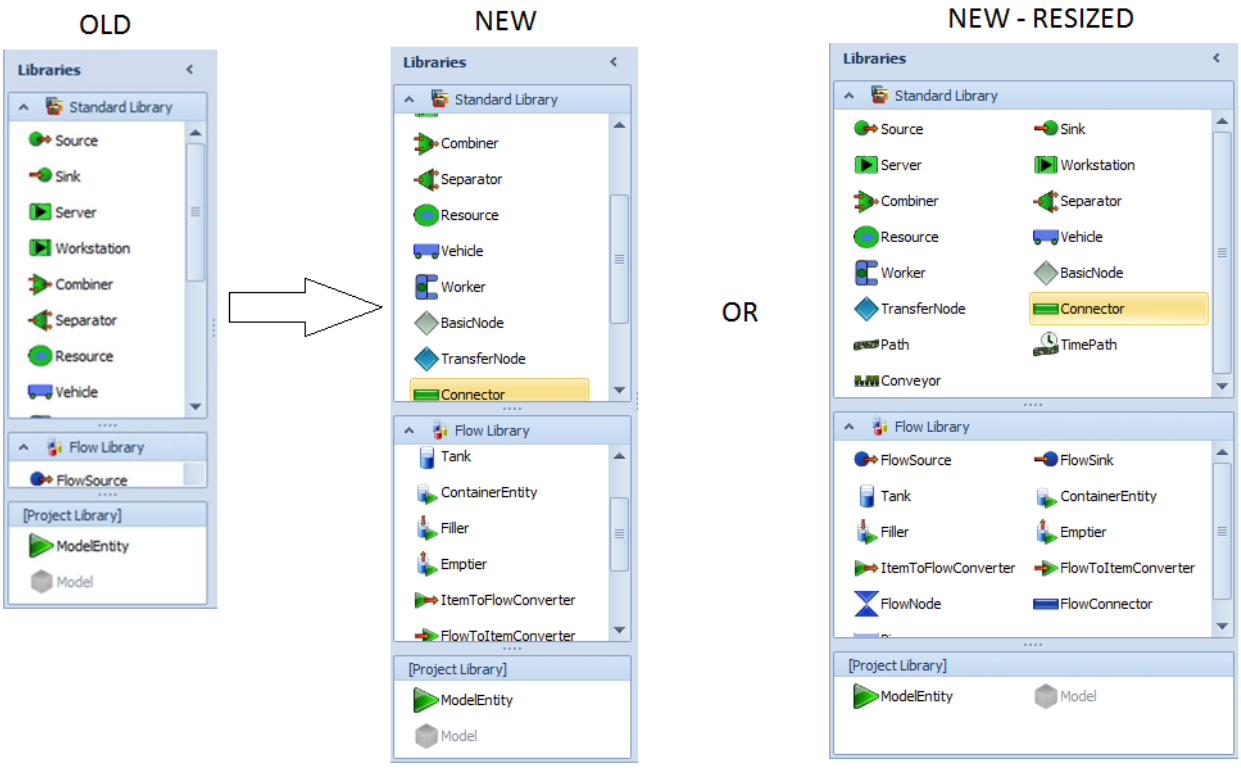
Table Usability Enhancement – Split

In conjunction with the table enhancements from the previous sprint, we now allow data tables to be 'split' for ease of use in viewing and editing large tables. The *Split* option is available by right-clicking anywhere in the table and selecting the option. The splitter bar down the middle of the table, as seen below, can be moved and each portion of the table view contains its own scroll bar along the bottom.



Library Panel Changes – Concurrent Active Libraries

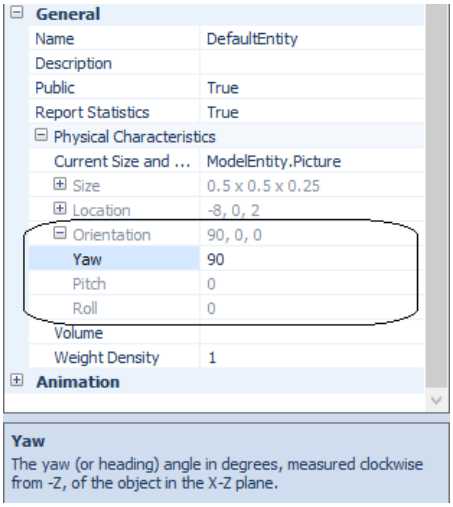
Our interface for the Libraries panel has been enhanced to allow users to easily work concurrently with multiple libraries. As you can see in the diagram below, the old panel allowed for a single library of objects to be active and selected at a time. The new panel interface allows all library panels to be active at once. The interface provides scroll bars for each library and users still expand/collapse or adjust the splitters for each library individually. Additionally, the side splitter can be dragged to get a resized Libraries panel as shown below.



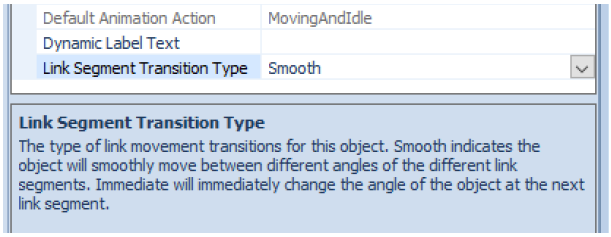
Animation Enhancements

There is a new Orientation property group under the Physical Characteristics category. The end user can adjust the *Yaw* value, but the *Pitch* and *Roll* values are always read-only and set to zero (not supported graphically right now). The *Yaw* (or heading) is defined as the angle in degrees, measured clockwise from –Z, of the object in the X-Z plane.

There is also a new function **Orientation.Yaw** which returns the value of the *Yaw* from the property grid for Fixed objects, and the value of *Movement.Heading* for Agent objects.



We have added a new property in the Animation group, available for entity instances, which will determine how entity objects move between angles of link segments:

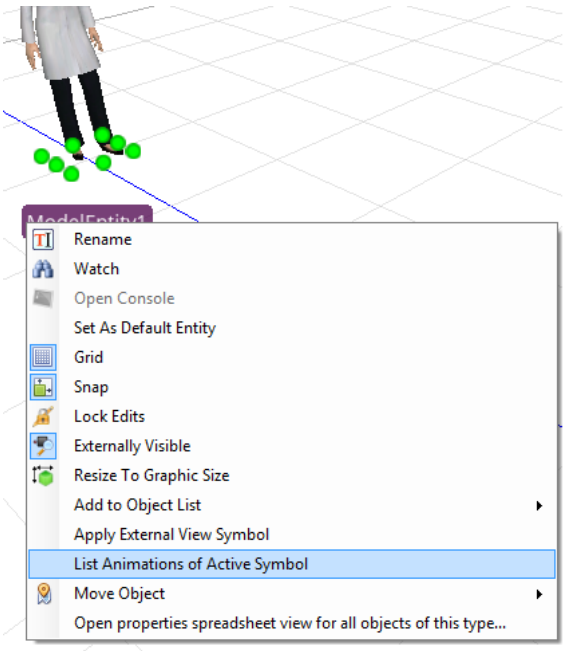


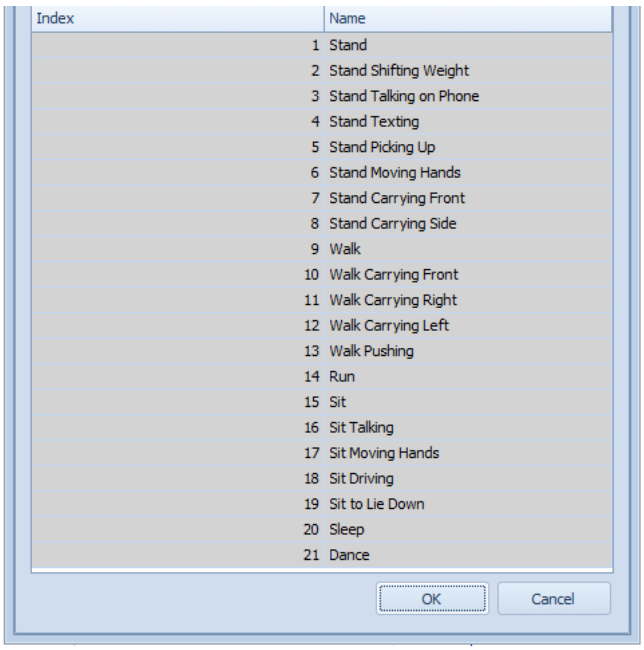
Simio Release 8 – Sprint 142 – July 18, 2016

In this sprint, we have made a number of user-requested enhancements and are continuing to work on more table and schedule type features for the next sprint.

Revised and Consistent Animated Symbols

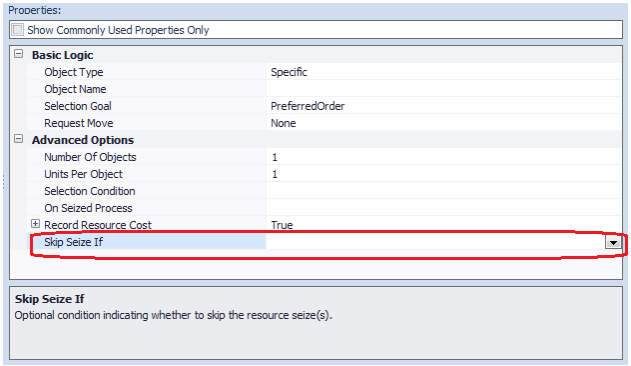
We have enhanced our animated symbols selection to include more realistic animated people symbols as well as a consistent set of 21 actions amongst all animated peoples. These include the actions as shown below.





Seize/Consume/Produce step Enhancements

The Seize, Consume and Produce steps now include an option to conditionally skip a seize attempt or consume / produce materials attempt. Note that corresponding enhancements to the Release step are not required because the 'All' *Release Quantity Type* will simply not release any resources when none had been seized by the executing token.



New Table Based Work Schedules

Schedules now allow for either Pattern Based Work Schedules (only option prior to Sprint 142) or Table Based Work Schedules (new).

The user interface for Pattern Based schedules has been updated slightly to show both the work schedule and day patterns within the same window, as shown below.

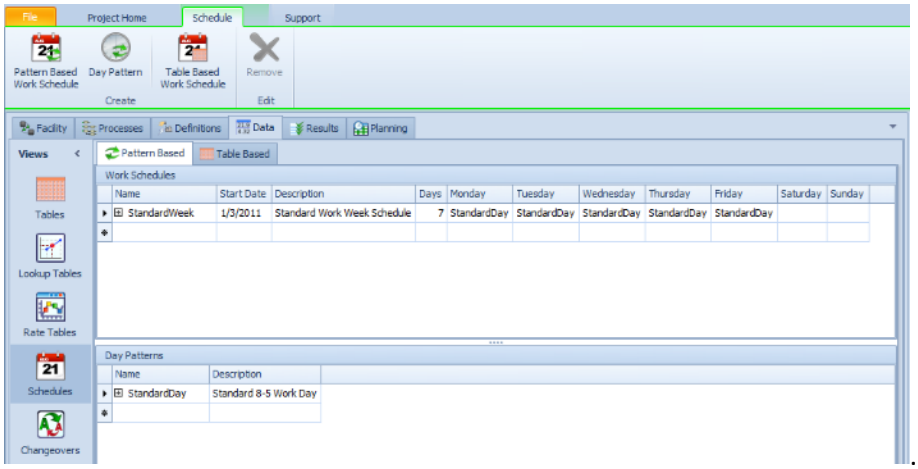


Table Based schedules require the schedule to reference a Table name, as well as property values for *Start Date Time*, *End Date Time*, *Value* and *Cost Multiplier*, as shown below. These table based schedules allow for users to easily import data into the tables from external data sources. Note that *Value* and *Cost Multiplier* expressions are evaluated only at the <u>start</u> of the simulation run and are not dynamically updated.

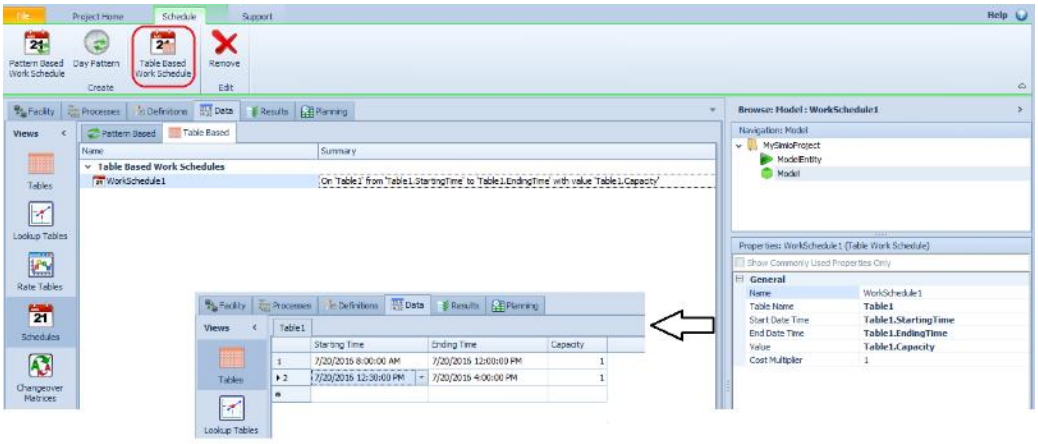
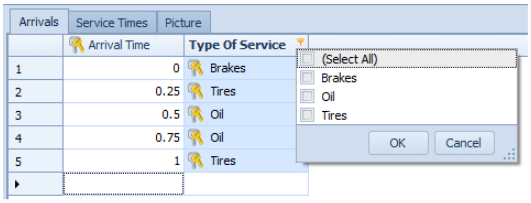
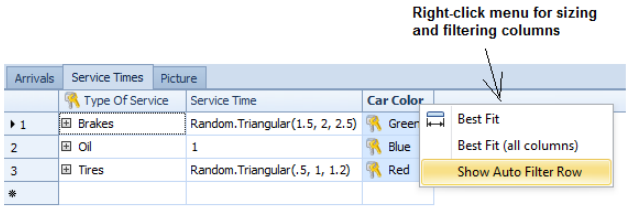
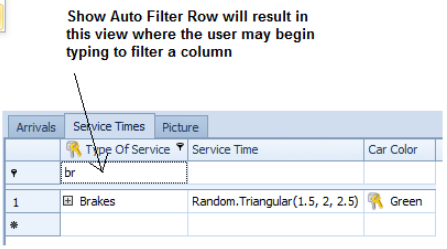


Table Usability Enhancements

We are continuing to add enhancements to our table usability. This sprint we have included the multi-select filter on columns, as well as the auto filter row that can be turned on for typing in column data to filter.





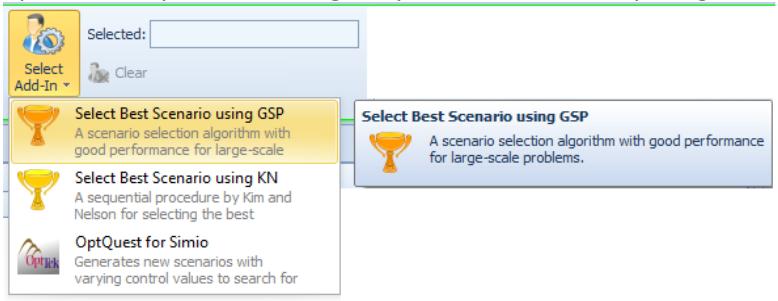


Support Distributed Runs for Academic Enterprise Edition

We now support distributed runs for the Simio Enterprise Edition – Academic version for experimentation.

New Experiment Add-In

We have added an experiment add-in for selecting the best scenario for large scale models. Select Best Scenario using GSP add-in uses the 'Good Selection Procedure' to solve large-scale simulation-based optimization problems using the power of cloud computing.

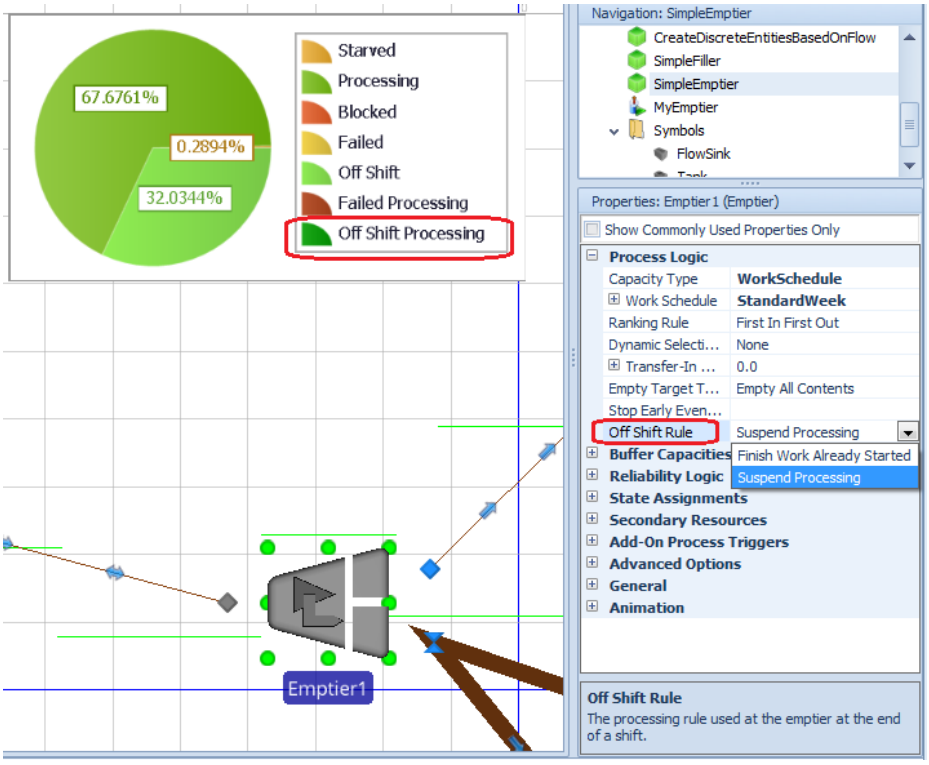


Simio Release 8 – Sprint 140/141 – June 27, 2016

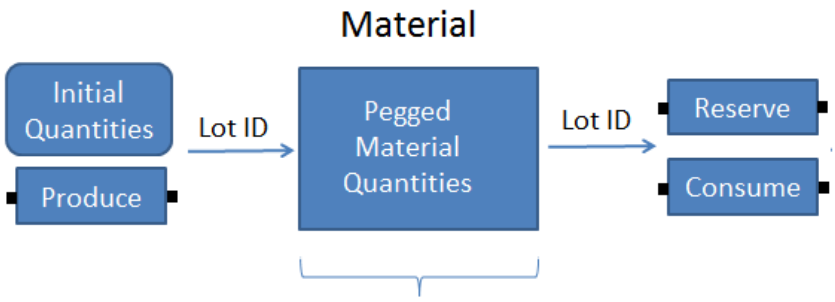
In this sprint, we have made a few advanced user requested features in addition to adding Material enhancements to the product. Both the concept of Material Pegging (using a Lot ID property) as well as Ignoring Material Constraints have been added to the Material element to better represent actual systems allocation and tracking of raw materials.

Flow Library – Filler/Emptier Object Enhancement - New *OffShift Rule* property
Last sprint, this new property was added to the Standard Library objects and we've now enhanced Filler and Emptier in the Flow Library objects to include this as well.

The new *OffShift Rule* property that allows either 'Suspend Processing' or 'Finish Work Already Started' at the end of a shift or capacity change. A corresponding new list state of 'OffShiftProcessing' (value of 6) has also been added to the corresponding objects. If the *OffShift Rule* is 'Suspend Processing', the processing of the entity(s) is immediately halted and the Filler/Emptier goes into an 'OffShift' state to complete processing once the capacity / shift changes. If the *OffShift Rule* is 'Finish Work Already Started', the state of the Filler/Emptier goes into 'OffShiftProcessing' while the processing of the entity(s) is completed, then will change to 'OffShift' for the remaining off shift time.



Material Pegging – Enhancement to Materials Element and Supporting Steps
Pegging links material demand to material supply. Simio will allow material quantities in stock to be assigned to lot identifiers, with those lot identifiers then referenceable by material reservation and consumption requests (see below).



Lot ID	Quantity In Stock
1	50
2	100
3	75

Pegging relationships between material demand and material supply such as shown above can provide more detailed logging of raw material usage as well as prevent material inventory from being allocated to unlinked demand.

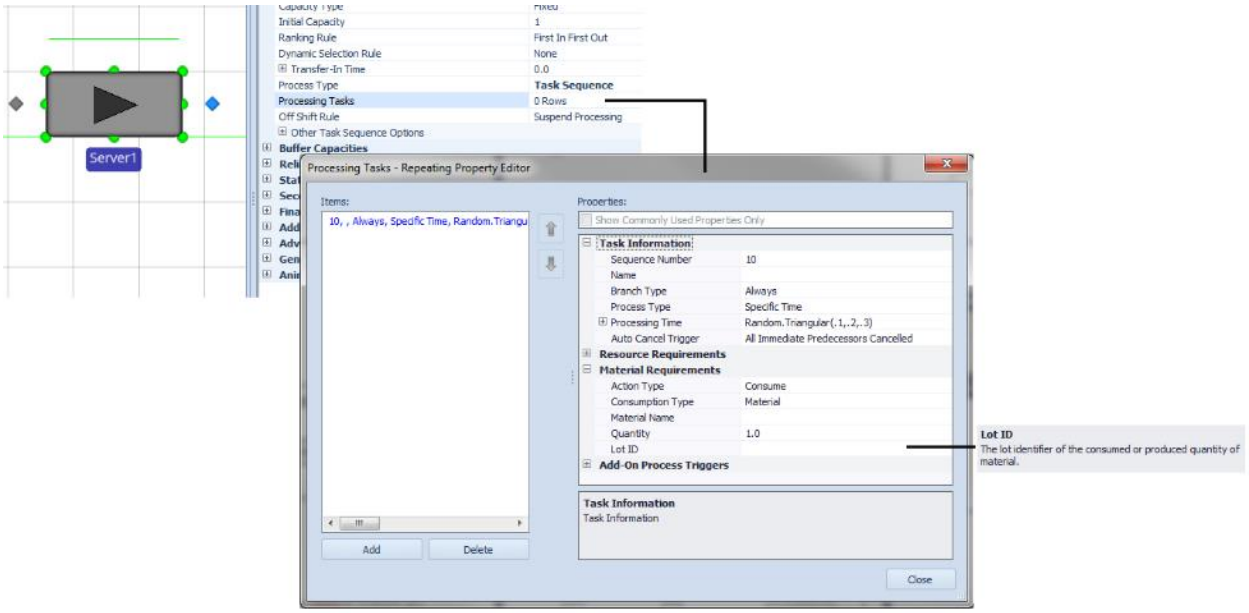
As an example, suppose a specific production order requires 100 units of raw material A. That material quantity can be linked (pegged) to the order using a unique lot number. Entities associated with other production orders will then be unable to reserve or consume from that specific lot of material.

The **Material** element has been enhanced to include a new *Lot ID* under the Initial Quantities (More) repeat group for initializing material lots. The **Produce** and **Consume** steps also include the new *Lot ID* for producing or consuming a particular lot. Similarly, the **Reserve** and **UnReserve** steps include the Lot ID property for reserving a particular lot of material. The **Route** step now allows the required materials before routing to include quantities that are assigned to specific lot identifiers.

The *Lot ID* property may be specified as a string literal enclosed in double quotes (e.g., “Lot1”), the name of a string state variable or string column in a table, or any other expression that returns a string value.

Material Pegging – Enhancement to Server, Combiner, Separator, Workstation and TransferNode
The Standard Library objects that include materials specification have been updated as well to include the above enhancements for *Lot ID*.

The **Server, Combiner,** & **Separator** objects have be enhanced with a new *Lot ID* property for defining the material requirements of a task sequence’s processing task, as shown in the diagram below. The **Workstation** object has been enhanced with a new *Lot ID* property for defining material constraints. The **TransferNode** object now includes a new *Lot ID* property for defining the required materials that must be available before routing out to a destination selected from a list.



New Functions for Material Pegging
The following functions have been added in support of the above material pegging enhancements.

Pegged.QuantityInStock(lotID*) - Returns the current quantity of the material of a specified lot identifier that is present in the system.

Pegged.QuantityReserved(lotID*) - Returns the current quantity of the material of a specified lot identifier that is reserved.

Pegged.QuantityReservedTo(owner, lotID*) - Returns the current quantity of the material of a specified lot identifier that is reserved for use by a specified owner object.

*Specifying the lotID argument as an empty string will return the current quantity of unassigned (not pegged) material that is present in the system or is reserved.

Materials Enhancement – Ignoring Material Constraints
The **Material** element has been enhanced to conditionally allow a material's availability to be ignored as a constraint. The *Assume Infinite Availability If* property is an optional condition that is evaluated whenever an entity is attempting to consume the material. If the condition is true, then the entity will be allowed to immediately consume its full required quantity of the material regardless of the actual quantity in stock. Note that this may result in negative inventory levels.

Materials Usage Log and Constraints Log Enhancements (Enterprise Edition) – Lot ID
The Material Usage Log now includes columns for Lot ID and Lot Stock Level, as shown below.

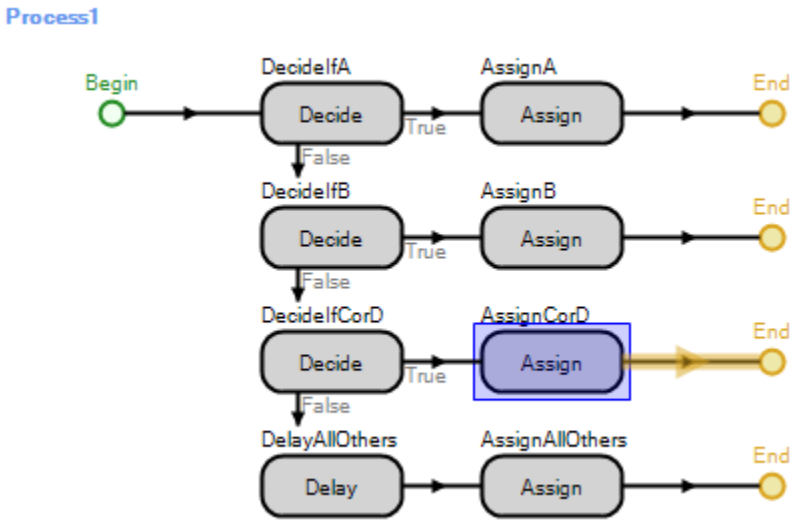
Time	Material	Lot Id	Entity Id	Entity	Quantity	Stock Level	Lot Stock Level
01/01/2008 5:00:00 AM	Widget				100	100	100
01/01/2008 7:00:00 AM	Widget	W13			40	140	40
01/01/2008 8:13:00 AM	Widget	W16			22	162	22
01/01/2008 9:45:00 AM	Widget				100	262	100
01/01/2008 4:00:00 PM	Widget	W16			-10	252	12

Note that the value shown in the existing Stock Level column hasn't changed; it continues to show the stock level for the material overall, while the new Lot Stock Level column shows the stock level only for the specific lot, or for the material that doesn't have a Lot ID.

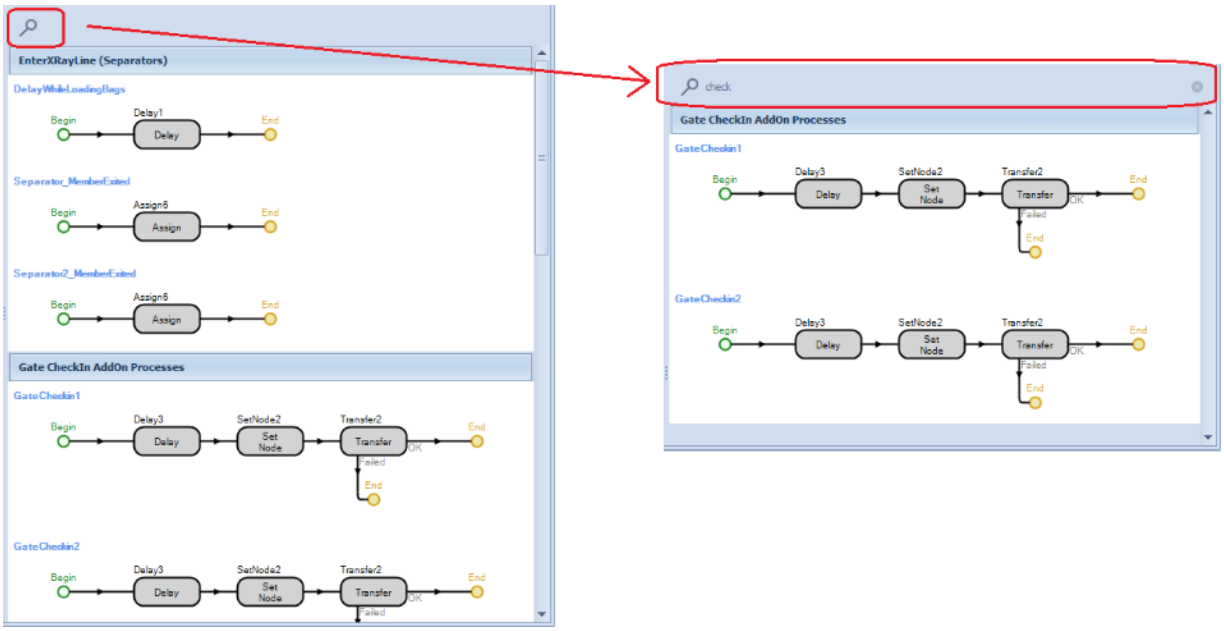
The Constraint Log now shows the Lot ID for material constraints with a lot specified. It appears as MaterialName.LotID in the Constraint Item column. Note that the Constraint Item Id column continues to show just the MaterialName.

Processes Window Enhancements

The Processes window has been changed to include a slightly new graphical interface for the process logic connections, as shown below.

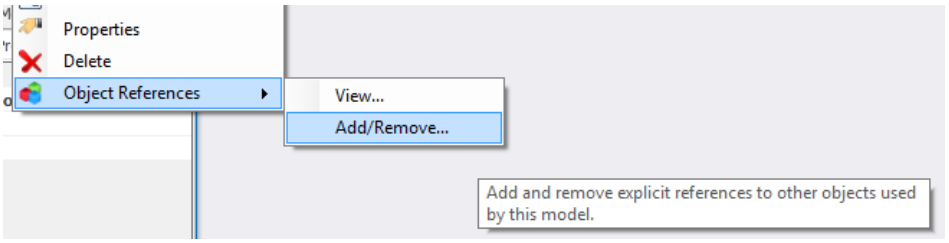


Additionally, there is new a Search filter mechanism for the processes themselves for easy locating by process name.



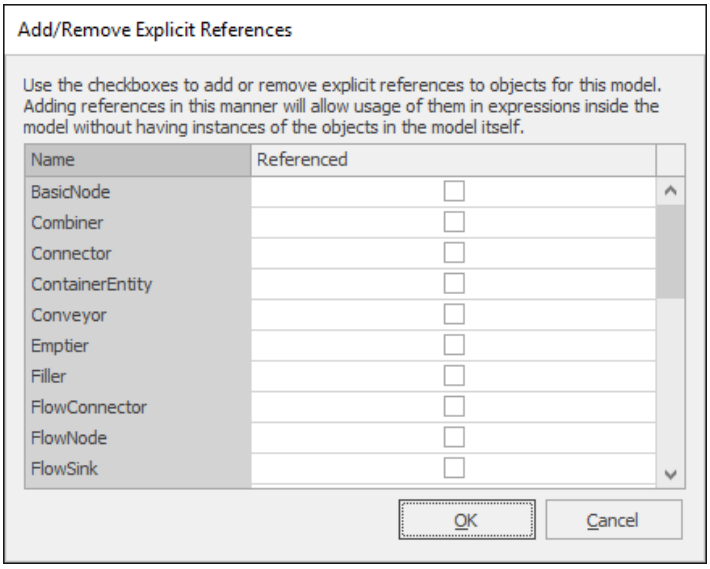
Model Object References

The right click menu on a Model has changed so that the Object Reference… selection is now a submenu:



The “View…” selection shows all object definitions referenced by the model, as this Object References option has in the past.

The “Add/Remove…” is new and allows a user to explicitly add a reference to another object definition, without having the instantiate something of that type inside the model. This is particularly useful if, for example, the user is creating a custom link that has StartingNode and EndingNode of a given type *Node. The user can then add a reference to *Node in the below dialog:



And then the expressions inside the link can reference “StartingNode.*Node.***”.

* Note that this is really only needed if referencing from other projects/libraries. Simio originally (and continues to) looks at sibling objects in the same project when resolving name and auto-referencing those.

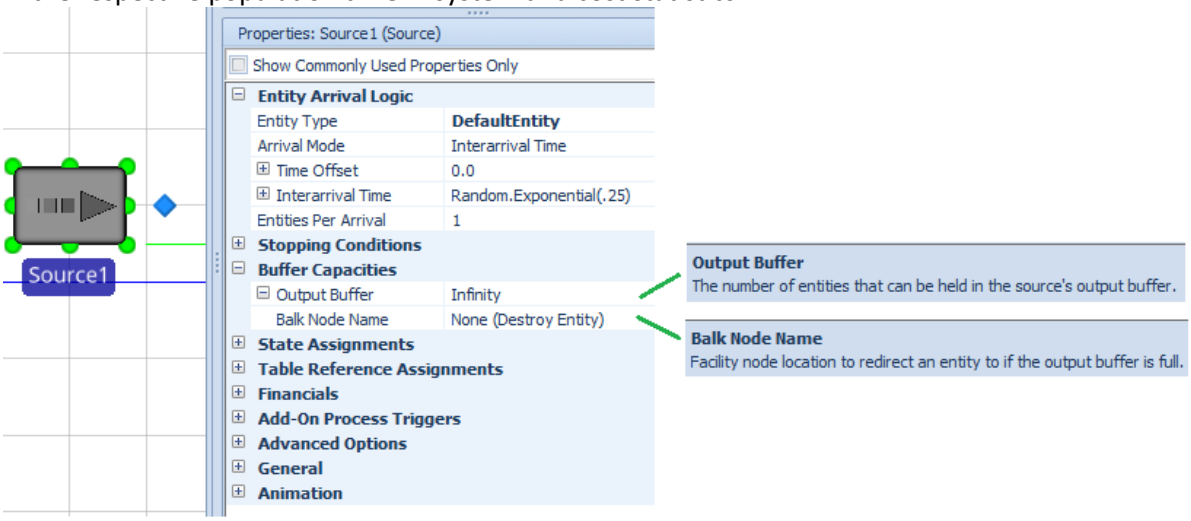
Simio Release 8 – Sprint 139 – May 20, 2016

In this sprint, we have added a number of enhancements to our existing object functionality that will make many things easier to do! These include balking from a Source, using off-shift rules for fixed objects, removing rows from output tables and logging of tasks within a task sequence.

Source Object - New Output Buffer Capacity & Balking Options

The Source object within the Standard Library now allows the user to specify a finite capacity for the output buffer. Entities may *balk* from entering a Source object's output buffer due to the buffer being full, allowing a user to either have those entities automatically destroyed or, alternatively, redirected to a specified node location.

Statistics are also included on the total number of balked entities in the automatically reported results. If, however, a Source object automatically destroys balked entities, then those entities are not recorded in the respective population time in system and cost statistics.



If entities have balked at entering a Source's output buffer, then the automatically reported results will include the following (shown also in the diagram below):

1. A 'NumberBalked' total will be reported by the output buffer station.
2. If there were balked entities automatically destroyed at the source, then those destroyed entities will not be recorded as observations in the population 'TimeInSystem' and 'CostPerItem' tallies for the entity type.

Average

Drop Column Fields Here

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
ModelEntity	DefaultEntity	[Population]	Content	NumberInSystem	Average	10.9674
					Maximum	12.0000
			Costs	CostPerItem	Average (USD)	1.0000
					Maximum (USD)	1.0000
					Minimum (USD)	1.0000
					Observations	143.0000
			FlowTime	TimeInSystem	Average (Hours)	1.7642
					Maximum (Hours)	1.8332
					Minimum (Hours)	0.1667
					Observations	143.0000
			Throughput	NumberCreated	Total	5,730.0000
				NumberDestroyed	Total	5,719.0000
Source	Source1	OutputBuffer	Content	NumberInStation	Average	9.9674
					Maximum	10.0000
			HoldingTime	TimeInStation	Average (Hours)	1.5980
					Maximum (Hours)	1.6665
					Minimum (Hours)	0.0000
			Throughput	NumberBalked	Total	5,576.0000
				NumberEntered	Total	154.0000
				NumberExited	Total	144.0000

2. Balked entities automatically destroyed at Source are not recorded in the population 'TimeInSystem' or 'CostPerItem' statistics.

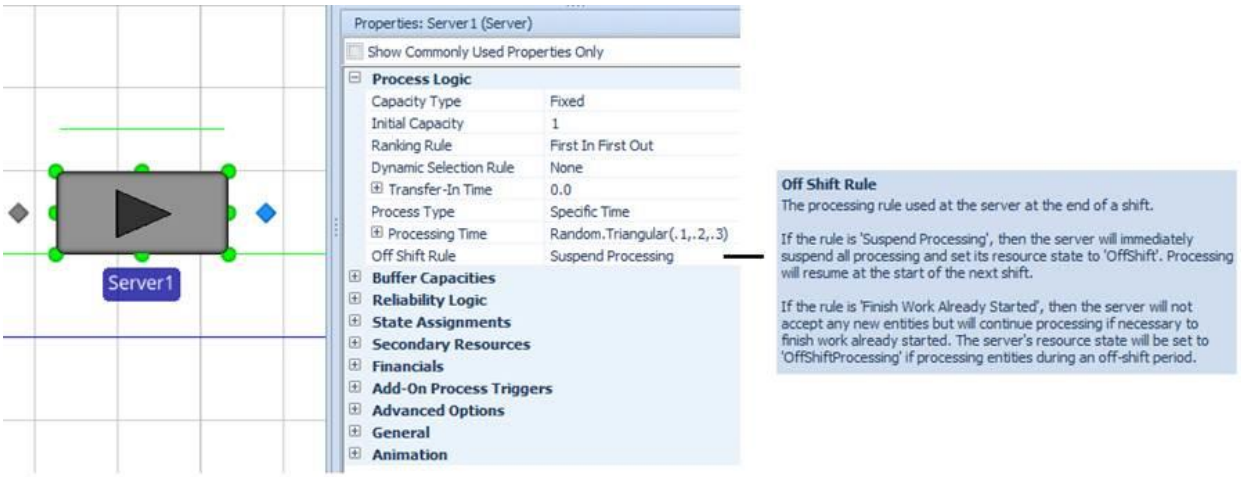
1. Total 'NumberBalked' Statistic

Server/Combiner/Separator Object Enhancement - New *OffShift Rule* property

We have added a new *OffShift Rule* property that allows either 'Suspend Processing' or 'Finish Work Already Started' at the end of a shift or capacity change. A corresponding new list state of 'OffShiftProcessing' (value of 6) has also been added to the corresponding objects.

If the *OffShift Rule* is 'Suspend Processing', the processing of the entity(s) is immediately halted and the Server/Combiner/Separator goes into an 'OffShift' state to complete processing once the capacity / shift changes.

If the *OffShift Rule* is 'Finish Work Already Started', the state of the Server/Combiner/Separator goes into 'OffShiftProcessing' while the processing of the entity(s) is completed, then will change to 'OffShift' for the remaining off shift time.

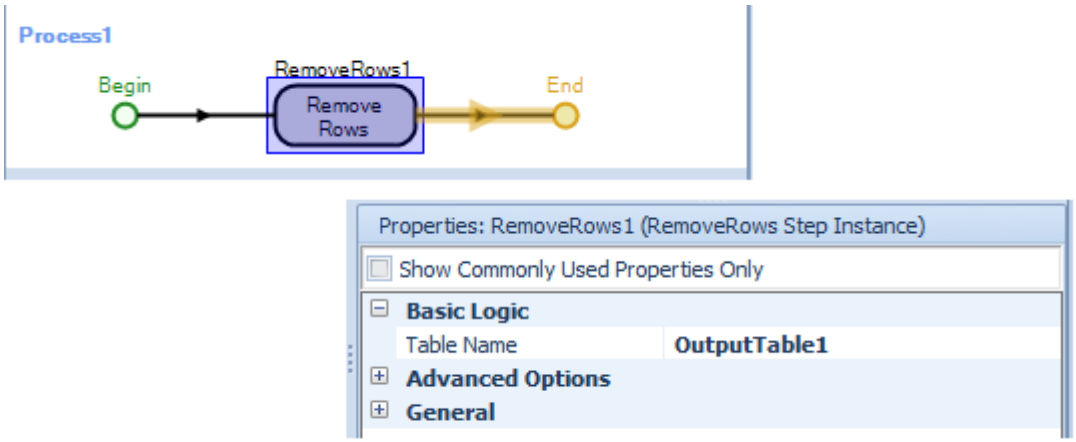


New String Functions

There are several new string functions that have been added to the software per customer request:

String.ToLower() - Returns a copy of a specified string converted to uppercase.
String.ToUpper() - Returns a copy of a specified string converted to lowercase.
String.Trim() - Returns a specified string with any leading and trailing white-space characters removed.
String.NewLine - Returns the newline constant defined for the local platform. Useful for adding a line break to form a multi-line string.

New RemoveRows Step
We have added a RemoveRows step that will allow users to remove the all current rows within an output table. Any token or object references to that table will become invalid.

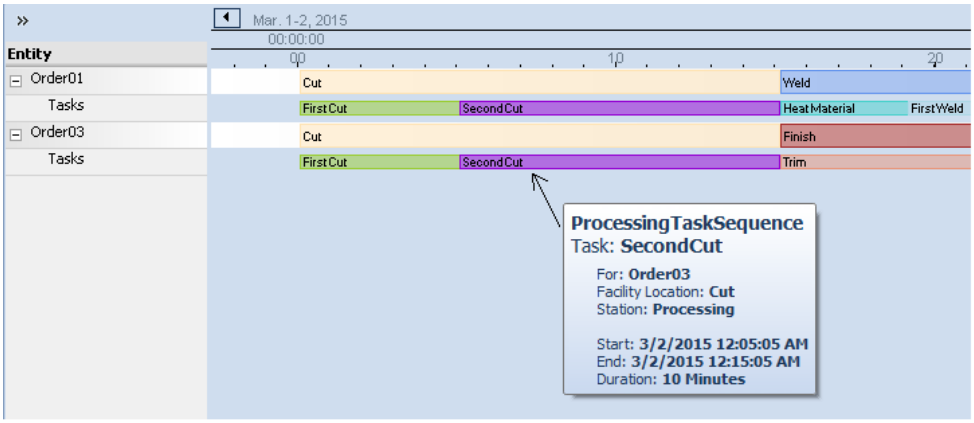


Task Sequences Enhancement – Task Log / Gantt (Enterprise Edition)
We have added an additional log to the Logs within Enterprise Edition that details information regarding the Task Sequences tasks specified. Task Sequence type processing is available through several of the Standard Library Objects such as Server, Combiner and Separator. Information within the task log include the entity, facility location (typically name of object), task, start time, end time and duration.

Resource Usage Log | Resource State Log | Resource Capacity Log | Constraint Log | Transporter Usage Log | Material Usage Log | Task Log

Entity Id	Entity	Facility Location	Station	Task Sequence	Task	Start Time	End Time	Duration (Hours)
MildSick.29	MildSick	Registration	Processing	ProcessingTaskSequence	FindPatientName	3/2/2015 12:00:05 AM	3/2/2015 12:05:05 AM	0.0833333
MildSick.29	MildSick	Registration	Processing	ProcessingTaskSequence	SendInfoToNurse	3/2/2015 12:05:05 AM	3/2/2015 12:15:05 AM	0.166667
MildSick.29	MildSick	BasicEval	Processing	ProcessingTaskSequence	GetVitals	3/2/2015 12:15:08 AM	3/2/2015 12:19:08 AM	0.0666667
MildSick.29	MildSick	BasicEval	Processing	ProcessingTaskSequence	EvalWithDoctor	3/2/2015 12:19:08 AM	3/2/2015 12:25:08 AM	0.1
MildSick.29	MildSick	BasicEval	Processing	ProcessingTaskSequence	FollowupTask	3/2/2015 12:25:08 AM	3/2/2015 12:27:08 AM	0.0333333

In addition to the above log, within both the Resource Gantt and Entity Workflow Gantt, the Tasks for a resource can be shown by expanding the entity or resource using the '+' and turning on the Tasks within the Gantt ribbon Visibility section.

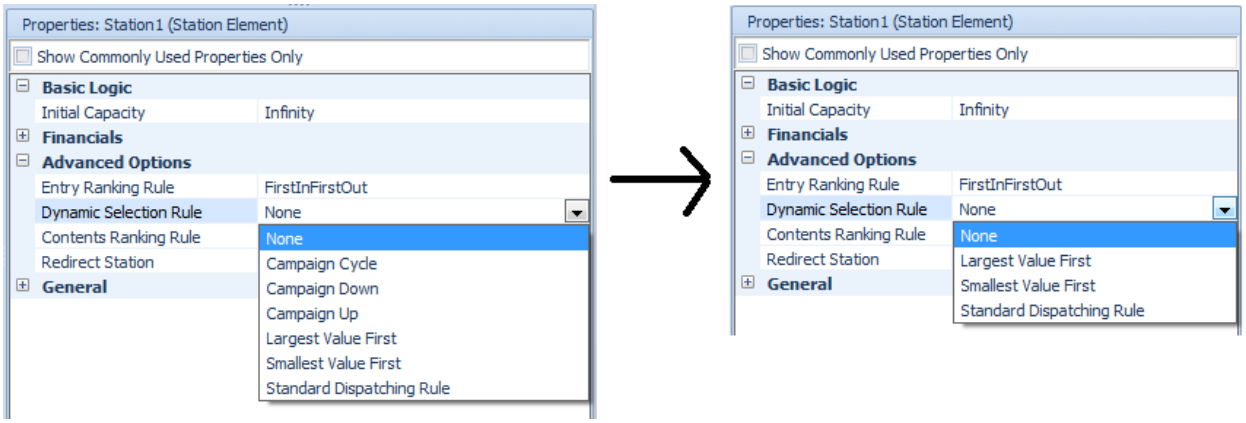

Mar. 1-2, 2015
00:00:00
Entity
Order01
Tasks
Order03
Tasks
Cut
Weld
First Cut
Second Cut
Heat Material
First Weld
Cut
Finish
First Cut
Second Cut
Trim
ProcessingTaskSequence
Task: SecondCut
For: Order03
Facility Location: Cut
Station: Processing
Start: 3/2/2015 12:05:05 AM
End: 3/2/2015 12:15:05 AM
Duration: 10 Minutes

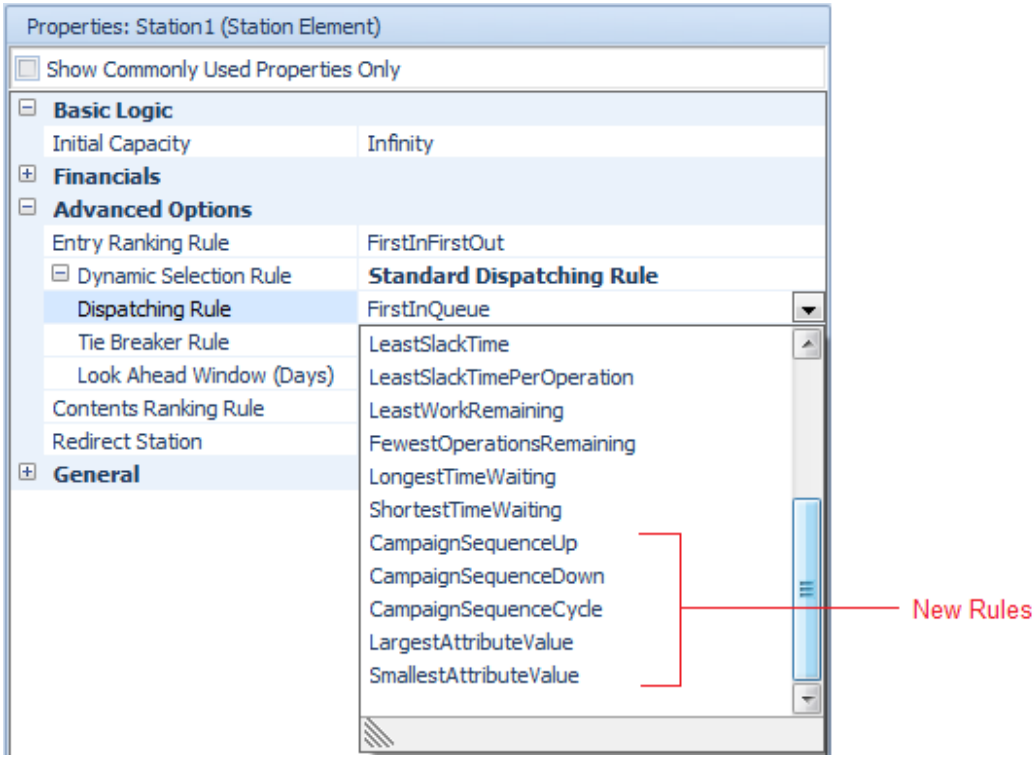
Simio Release 8 – Sprint 138 – April 25, 2016

In this sprint, we have added some user-requested features, such as completing the dynamic selection rule changes started in sprint 137, enhancing the Notify step and process properties within the Processes window and adding the frequently requested list reference states.

Dynamic Selection Rules – Simplified/Enhanced

The *Dynamic Selection Rule* properties are found in Server, Workstation, Worker, Vehicle, Resource, and TransferNode objects in the Standard Library objects. They have been changed from the original rules (see below left side) to the simplified rules (below right side) where the Standard Dispatching Rule list now includes an assortment of rules that can be used for both the *Dispatching Rule* and *Tie Breaker Rule* properties.

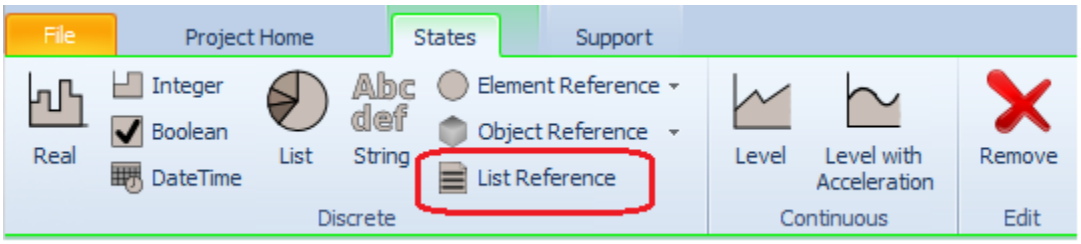




New List Reference States

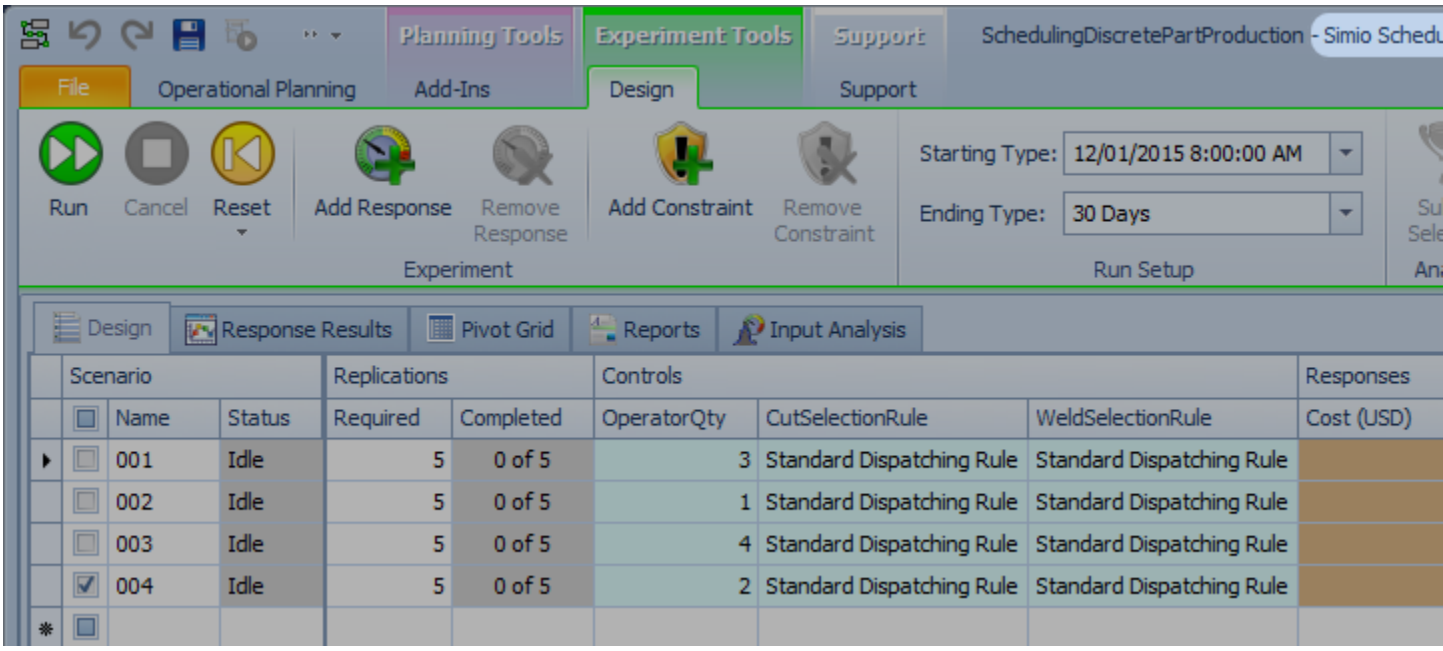
Within the States view, the Tables view (on the States ribbon), and the Tokens view, the List Reference state is now also available. The List Reference state variable defines a list reference variable that may be changed by assignment logic at discrete times during a model run. You can assign a value to a list reference state using an Assign step in a process. The expression used as the value for the assignment must return a list reference, for example from an Object List property or another List Reference State.

This may be useful for assigning a different list to different entity types, for example, using something like ModelEntity.MyWorkerList, where MyWorkerList is a List Reference state on the ModelEntity object.



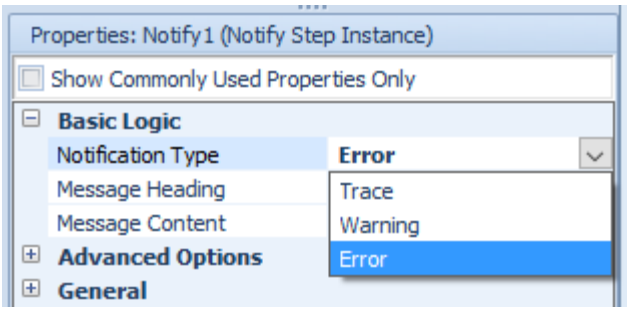
Scheduling Edition Enhancement

A Scheduling user can now see existing experiments, add or remove scenarios, targets, and constraints, run the experiments and view results.

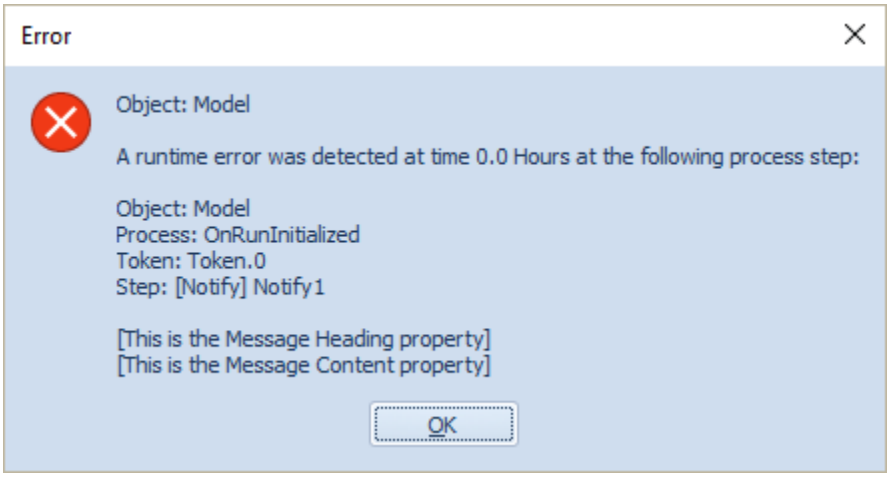


Notify Step Enhancement

The *Notification Type* property on the Notify step now includes the option to have an error and end the simulation run.

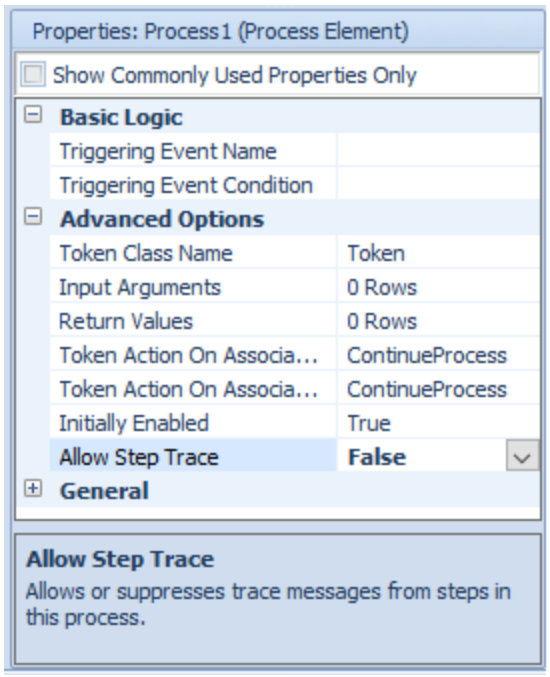


When the step is executed, the following error box will appear with the *Message Heading* and *Message Content* properties displayed in the error dialog.



Trace On/Off within Process

Within the properties window of each process in the Processes window, the *Allow Step Trace* property determines whether trace will be displayed for that process when the simulation model is run. The default value is 'True'. When specified as 'False', steps in this process will not produce trace messages. Note that this applies ONLY to steps in this process; it does not propagate to steps in other processes that might be called from this one.


Properties: Process1 (Process Element)
Show Commonly Used Properties Only
Basic Logic
Triggering Event Name
Triggering Event Condition
Advanced Options
Token Class Name
Token
Input Arguments
0 Rows
Return Values
0 Rows
Token Action On Associa...
ContinueProcess
Token Action On Associa...
ContinueProcess
Initially Enabled
True
Allow Step Trace
False
General
Allow Step Trace
Allows or suppresses trace messages from steps in this process.

Simio Release 8 – Sprint 137 – April 4, 2016

In this sprint, we have added some user-requested features, such as enhancements to the popular Task Sequences. Tasks can now be specified with predecessor or successor tasks, making parallel task diagrams much easier to specify. We've also updated the Flow Library converter type objects with a user-requested feature.

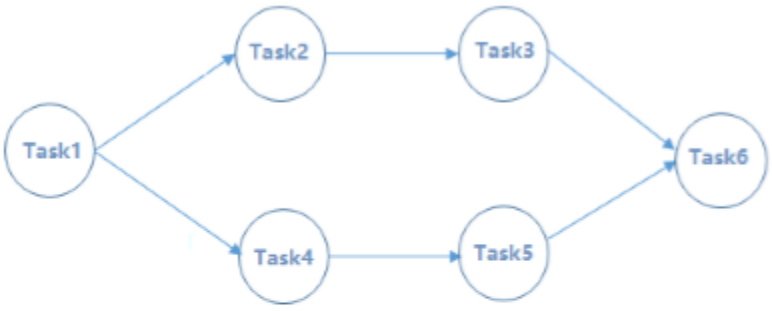
Task Sequences Enhancements

Task Sequence Element Changes

Within the Task Sequences element, we have added a Task Precedence Method property that provides three (3) alternative methods for defining the task precedence constraints that determine a task sequence:

- Using task sequence numbers (SequenceNumberMethod),
- Specifying the immediate predecessors for each task (ImmediatePredecessorsMethod), or
- Specifying the immediate successors for each task (ImmediateSuccessorsMethod).

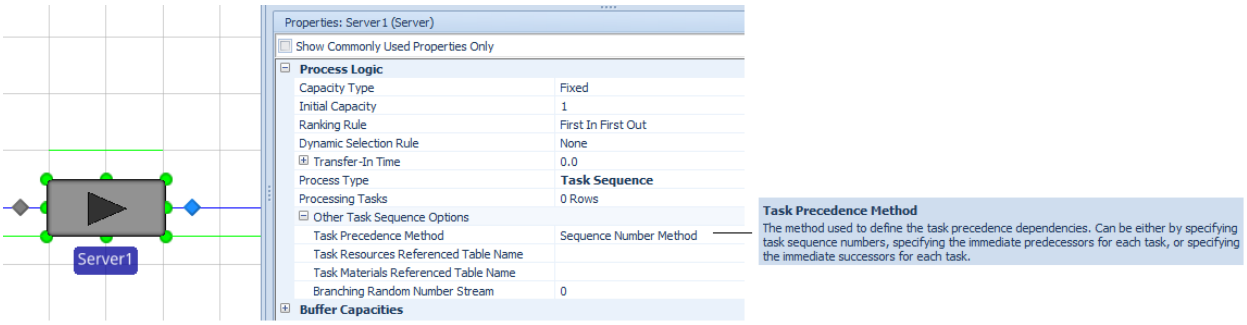
With predecessors or successors method, an *ID Number* is used instead of a *Sequence Number*. Thus, in addition to the prior SequenceNumberMethod defining the Sequence Numbers as 10, 20.1, 20.2, 30.1, 30.2 and 40, EITHER the immediate predecessors or immediate successors can be specified, as shown below.



ID Number	Task Name	Immediate Predecessors	Immediate Successors
1	Task1		2,4
2	Task2	1	3
3	Task3	2	6
4	Task4	1	5
5	Task5	4	6
6	Task6	3,5	

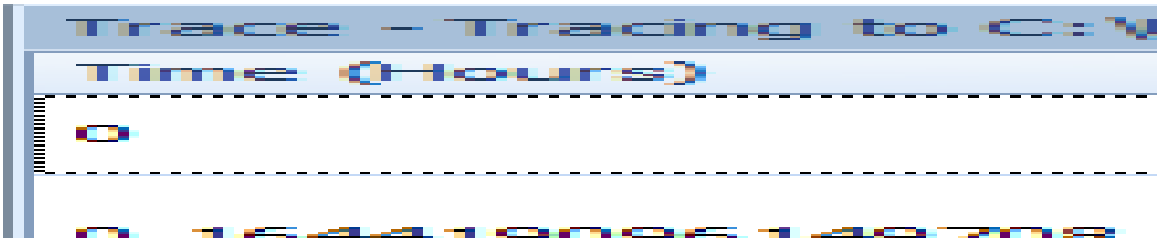
Server, Combiner, Separator – Task Sequence Enhancements

The above enhancements have also been incorporated into the Server, Combiner and Separator objects which utilize the 'Task Sequence' option for *Process Type*. The Other Task Sequence Options section of properties for these objects now contains the *Task Precedence Method* property, while the Processing Tasks repeat group contains the *ID Number* and either *Immediate Predecessors* or *Immediate Successors* properties.



Task Sequence Trace

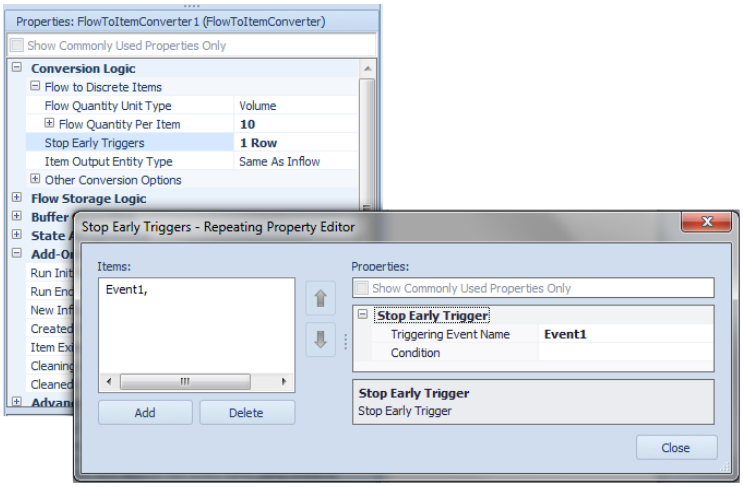
The model trace when using Task Sequences within the Standard Library objects, such as Server, has been enhanced. The trace is now easier to filter and see when each task has started and finished within an object, as shown below. For example, a user may filter the trace window by a specific Entity and specific Object (i.e., Server). Then the Step column can be filtered to '[StartTasks] ProcessingTasks' to see the trace messages specifically related to the task sequence execution by a specific entity at a specific server.



Flow-Related Enhancements

Converter Changes

The FlowToItemConverter and ItemToFlowConverter objects have been changed to include multiple *Stop Early Triggers* events. These optional event-driven triggers will immediately cause the FlowToItemConverter to stop waiting for additional inflow and create a new discrete entity early, while causing the ItemToFlowConverter to stop producing outflow for the current discrete entity early.



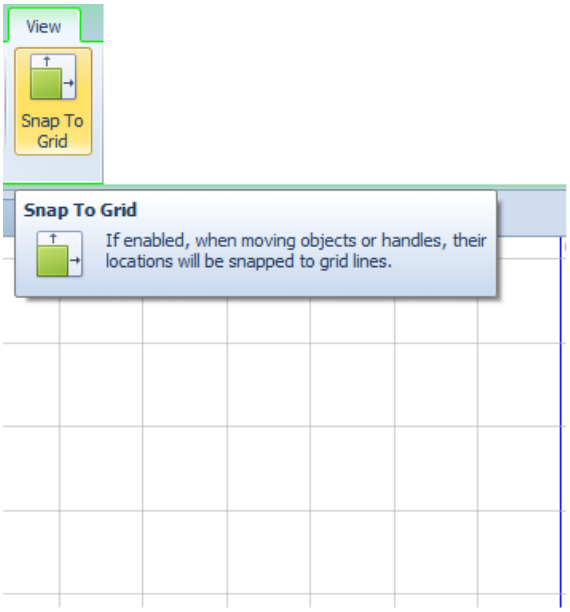
Simio Release 8 – Sprint 136 – March 16, 2016

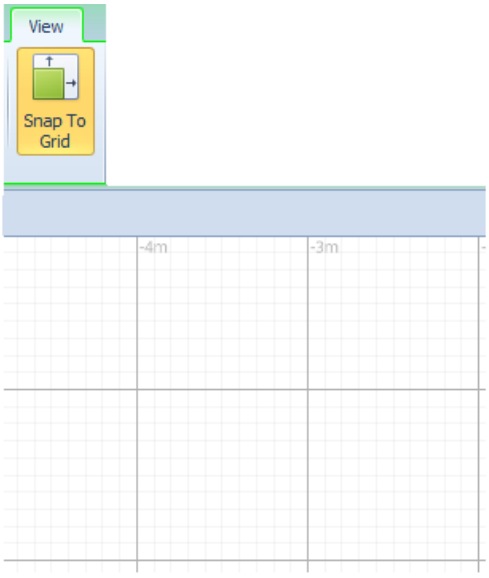
In this sprint, we are excited to add the much requested Snap to Grid feature to Simio to allow for more precise object alignment and rotation, as well as link placement. We've also expanded upon our mapping features, now providing options to move objects to particular map locations and draw links along road map routes.

Snap to Grid

A Snap To Grid button has been added to the View ribbon and will initially be enabled for new models (disabled for existing models prior to sprint 136).

Snap to Grid will snap moving object or handle locations to grid lines to allow for object alignment. The snap locations will be dependent on the grid level. In the below screen shot, the left grid is the default zoom location and objects will snap to those larger grid lines when placed. The right grid shown is zoomed in several times and displays more granular grid lines, in which case, the objects will snap to those smaller grid lines, allowing more precise placement.

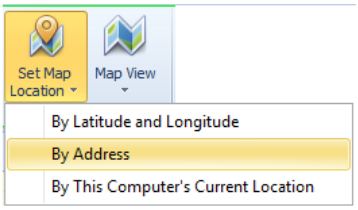


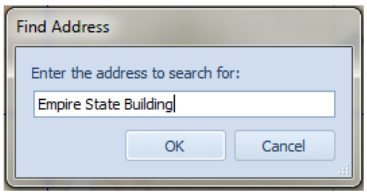


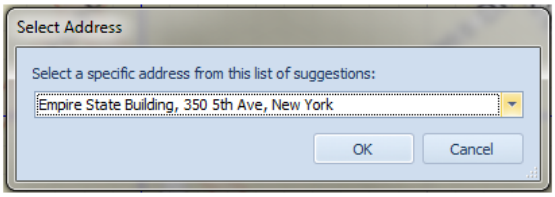
Additionally, Snap to Grid also snaps rotations of objects. When rotating objects with snap enabled, the object will rotate in 15 degree increments (thus 4 rotations to move the object 90 degrees).

Enhanced Map Support

We have expanded upon our map support that was added in sprint 135, including the ability to set the 0,0 coordinates to a specific location, move objects to particular specified locations and add links between specific nodes based on map road routes.







In the below example, nodes were placed at specific locations using the right-click Move Object / To Address option to the Empire State Building and Central Park, both in NYC. Then, by using Ctrl-click to selection both transfer nodes, a link can be placed between them using the 'Add link following map road route…' option. The link drawn then will have its *Drawn to Scale* property set to 'False' and its Logical Length set to the measured length of the entire route.

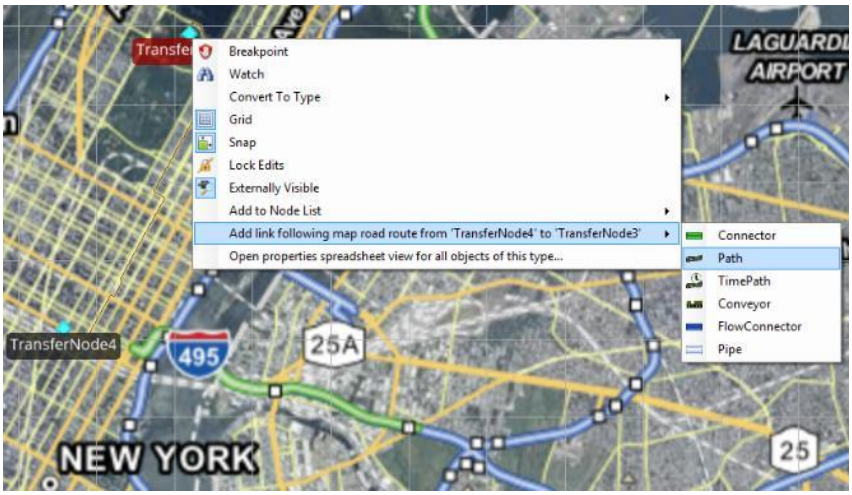
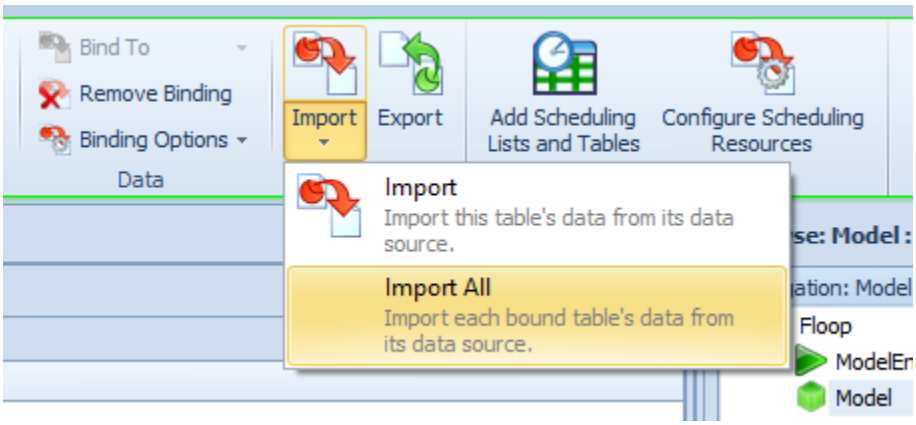


Table Importing Enhancements

Importing of tables has become easier by providing two options for Import including the import of the current table (only functionality prior to sprint 136), as well as import of all tables. The Import All will bring in all data from those that are bound to an external data source.



Additionally, for Excel imports, when importing large Excel files into Simio tables, we only re-load the Excel file if its timestamp indicates that the contents have changed since the last time it was loaded.

Enhanced Suspend and Resume Steps

Both the Suspend and Resume steps have been enhanced to allow multiple processes, object movements and/or regulatorflow to be suspended and/or resumed by using a single step. Repeatable properties of suspend actions have been added to the Suspend step and similar repeatable properties of resume actions have been added to the Resume step.

Updated Examples

The three (3) scheduling type examples that are included with the Simio Examples have been updated to incorporate the latest Standard Dispatching Rules that were added in Sprint 135. These examples include SchedulingDiscretePartProduction, SchedulingBicycleAssembly and SchedulingBatchBeverageProduction.

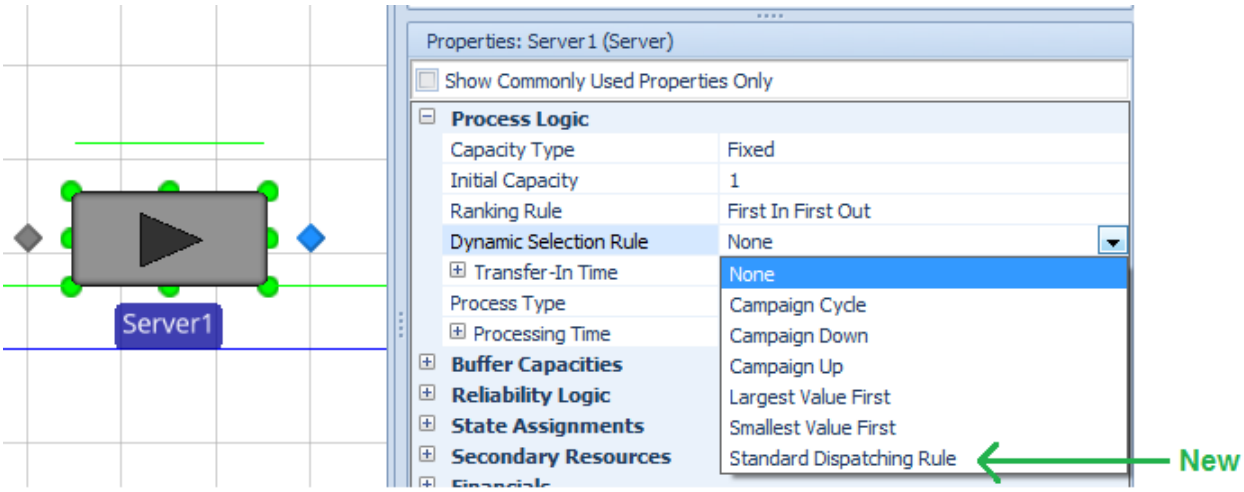
Simio Release 8 – Sprint 135 – February 23, 2016

In this sprint we have added quite a few user requested features, including standard dispatching rules for dynamic queue selection, as well as shift/ctrl-select for copy/paste of multiple processes.

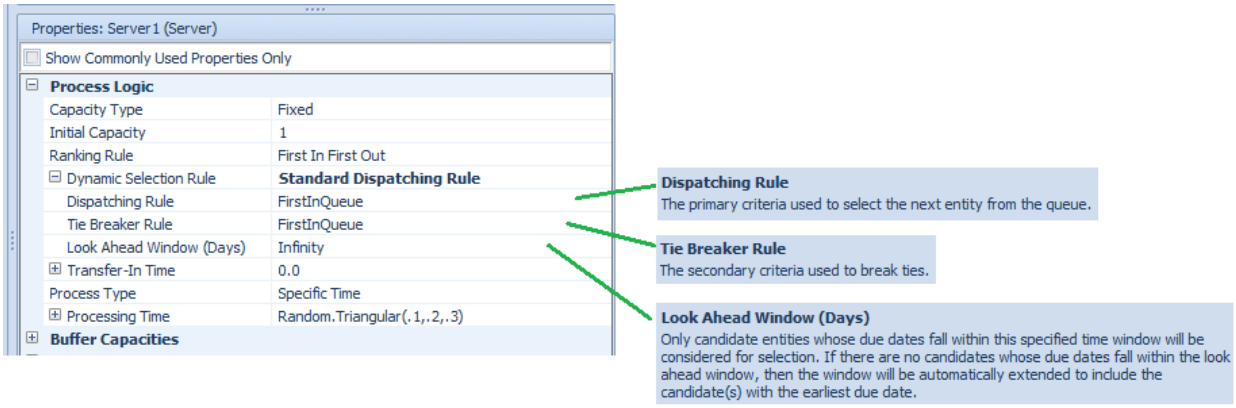
Standard Dispatching Rules

Standard Library Entity Object Enhancements

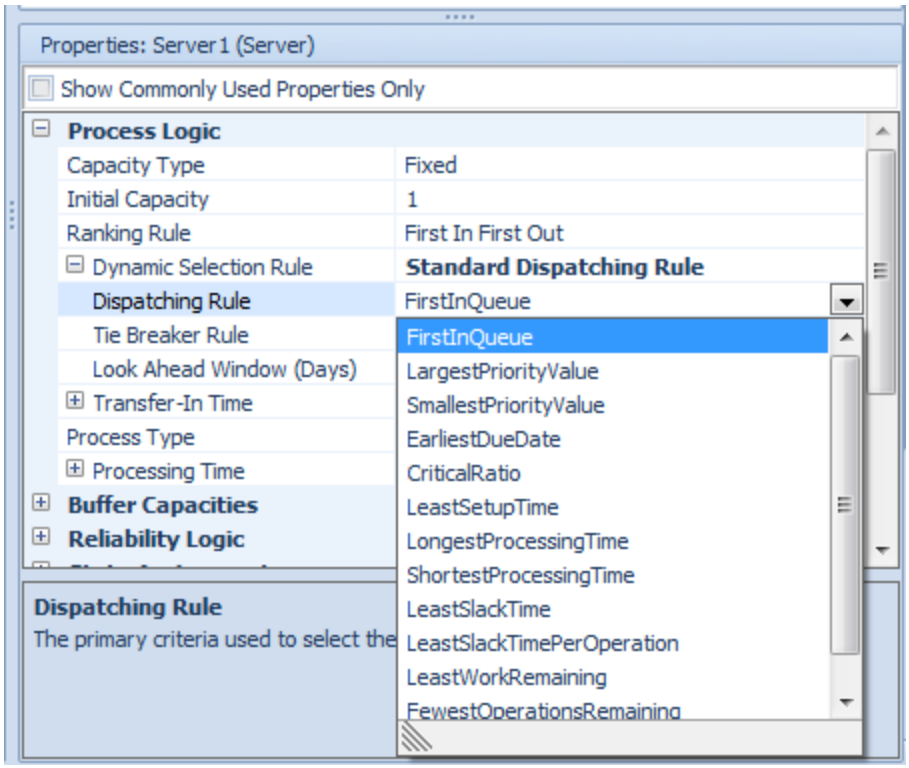
The Server, Combiner, Separator, Workstation, Resource, Worker, and Vehicle of the Standard Library and Filler and Emptier of the Flow Library have been updated with a new *Dynamic Selection Rule* option.



When the *Dynamic Selection Rule* is set to 'Standard Dispatching Rule', the user is then prompted for a *Dispatching Rule*, *Tie Breaker Rule* and *Look Ahead Window (Days)*.



Most of the standard dispatching rules to be supported are displayed below. There are currently fourteen (14) new rules that are supported – please refer to the Dynamic Selection Rules page within the help for more detailed information.



IMPORTANT NOTES: If you are opening an old model and want to use the new 'Standard Dispatching Rule' *Dynamic Selection Rule* with Standard Library objects (like Workstation), in that old model you need to do the following:

1) Go to the Run ribbon -> Advanced Options and click on the arrow in the bottom right corner to open Advanced Compatibility Settings. Set the *Always Associate Dynamic Selection Rule Evaluations With The Objects Making The Selections* setting to its default of 'True'. There was an issue in our expression evaluations in the previous software that had to be fixed to allow this new selection rule to work, but it is set to 'False' when opening old models to protect compatibility (just in case there is an old model assuming the behavior for some reason).
2) Go to Advanced Options of each Server or Workstation of interest, and right click on the *Expected Setup Time Expression* and *Expected Operation Time Expression* properties and reset the values to the default values.

Other Miscellaneous Supporting Properties and Functions

In support of the above features, we have added two new properties to the base fixed object, which includes Server, Combiner, Separator, and Workstation of the Standard Library and Filler and Emptier of the Flow Library.

ExpectedSetupTimeExpression – The expression used to estimate an expected setup time for an entity at a given fixed object.
ExpectedOperationTimeExpression – The expression used to estimate an expected operation time for an entity at a given fixed object.

In conjunction with the above properties, we have added functions for the same objects to include:

ExpectedSetupTimeFor(entity) – Returns an estimate of the expected setup time for a specified entity at the fixed object.
ExpectedOperationTimeFor(entity) – Returns an estimate of the expected operation time for a specified entity at the fixed object.

Additionally, the Activity element has been updated to include new functions (two of which replace current functions – see the Help for Obsolete Functions):

ExpectedTotalDurationFor(entity) – Returns an estimate of the expected total time duration (in hours) for the specified entity to perform the activity, calculated under the assumption that the entity would be next to start the activity.
NumberBatchesRequiredFor(entity) – Returns the total number of sequential batches required for the specified entity to perform the activity.
CurrentBatchDurationFor(entity) – Returns the time duration (in hours) to perform the activity for the specified entity's current batch. If the entity is not currently performing the activity then NaN is returned.

A new function for queue states includes:

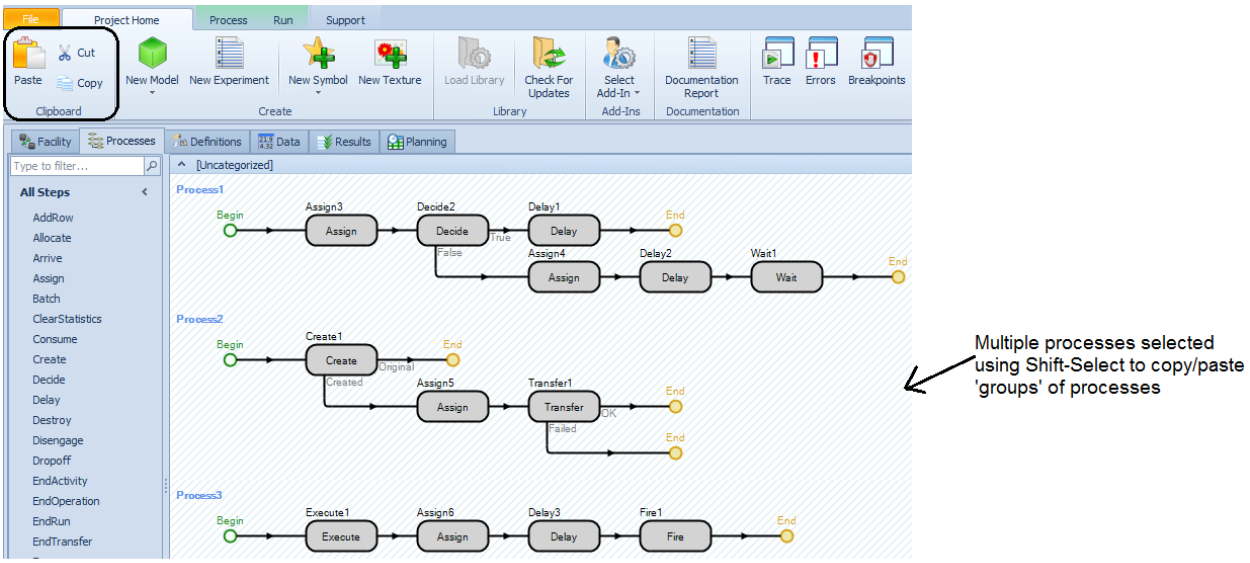
TimeWaiting (object) – Returns the time that a specified object has been waiting in the queue. If the queue does not contain the object then NaN is returned.

And finally, a new function for entity objects includes:

CurrentRoutingGroup – If the entity object is currently waiting in a route request queue, then this function returns a reference to the routing group being used. Otherwise, the Nothing keyword is returned.

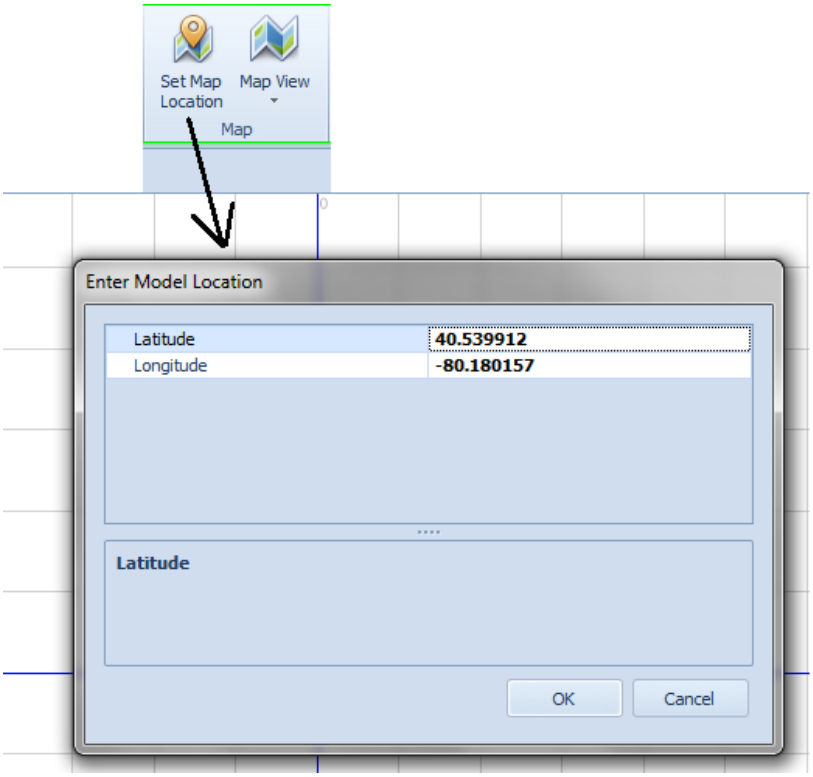
Shift-Select and Ctrl-Select within Processes Window

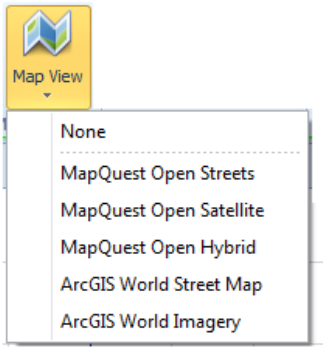
The Processes window now allows Shift-Click / Ctrl-Click to select multiple processes within the window for cut/copy/paste of multiple processes at the same time. When in the Processes window, click on a given process, then use the Shift key and click on another process below the already highlighted one – the first, last and all processes in between will be highlighted (see below). Additionally, using the Ctrl key and click on a process will continue to highlight selected processes (not in any necessary order). The Ctrl-X/Ctrl-C/Ctrl-V can then be used to cut/copy/paste these multiple processes. Alternatively, the Cut/Copy/Paste buttons on the Project Home ribbon can be used.



Background Map Support

Within the View ribbon, there are now two new icons under the Map grouping, including Set Map Location and Map View. Set Map Location allows the user to specify the latitude and longitude values that will correspond to the 0,0 location within the Facility window. The Map View option allows for pulling in various internet maps as background into the Facility window.





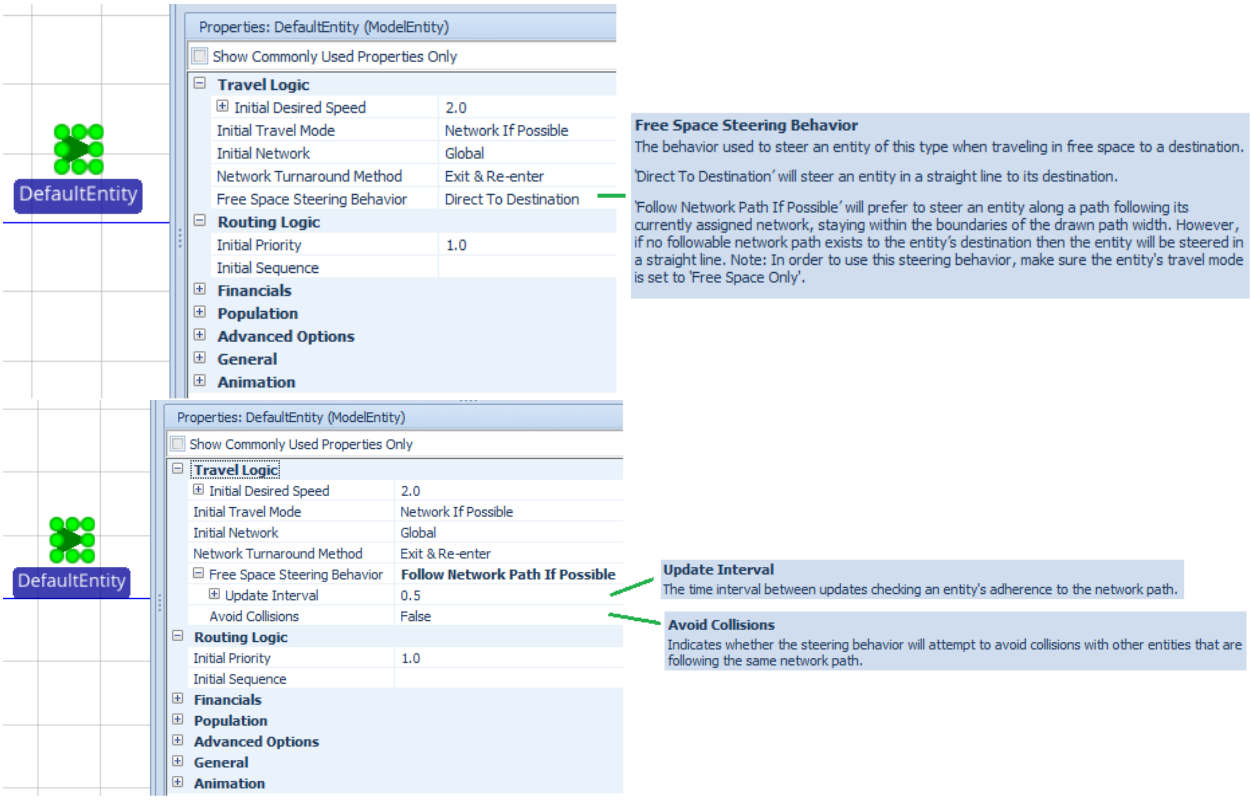
Simio Release 8 – Sprint 134 – February 3, 2016

In this sprint we have continued with a number of entity travel related enhancements, including the ability to have entity type objects easily move in free space along a network path, simulating pedestrian traffic movement. We have also enhanced the Navigation window, where the project models and experiments are located, to allow users to re-order and collapse/expand various items for more organized projects.

Entity Travel Related Enhancements

Standard Library Entity Object Enhancements

The ModelEntity, Worker, and Vehicle of the Standard Library and ContainerEntity of the Flow Library are all considered 'Entity' type objects. Within each of these objects, we have added a new *Free Space Steering Behavior* property, as well as *Update Interval* and *Avoid Collision* supporting properties if the new steering behavior property is set to 'Follow Network Path If Possible'.



The *Free Space Steering Behavior* property is the behavior that is used to steer an entity type object when traveling in free space to a destination. If the property value is 'Direct To Destination' the entity will move in a straight line to its destination. If the property value is 'Follow Network Path If Possible' then the entity will be steered along a path following its currently assigned network, staying within the boundaries of the drawn path width. The entity's travel mode should be set to 'Free Space Only'.

It is important to note that the *Update Interval* can be adjusted based on the speed of the entity to assure the entity's adherence to the network path.

SetNode Step Enhancement

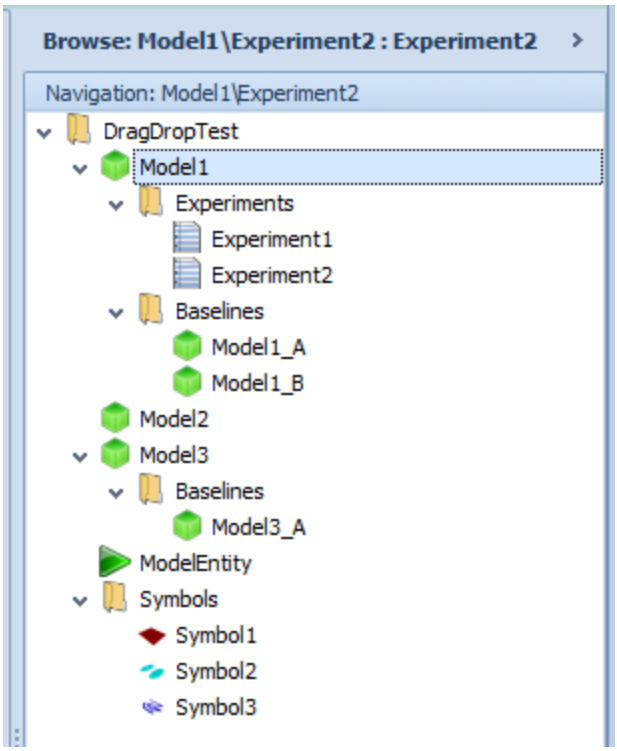
The SetNode step has an additional property, *Auto Clear If Visit Different Node*, which indicates whether to automatically clear the assigned destination node if for any reason the entity visits a different node beforehand. This enhancement has primarily been made to support the free space enhancement described above.

New Node Function

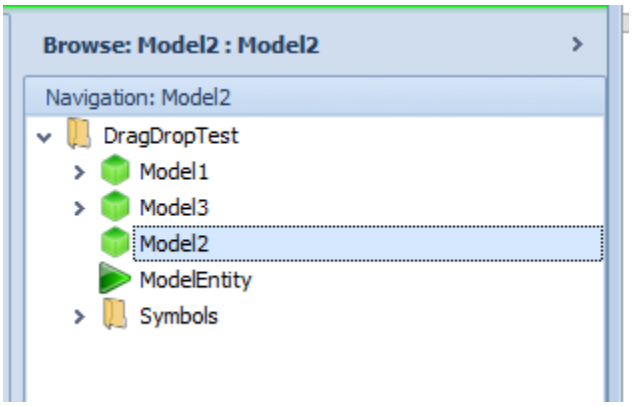
A new node function, available for BasicNode and TransferNode, has been added to the software, **OutboundLinks.SelectItemFor(entity)**. For a specified entity, this function returns a reference to a drawn outbound link from the node using the node's outbound link selection-related properties to make the selection decision. The OutboundLinks.SelectItemFor(entity) function will use the same outbound link selection logic that would be used if the entity were located at the node and executing a discrete Transfer from 'CurrentNode' to 'OutboundLink'. If the link decision is probability-based, then the value returned may not necessarily be the same as the link that the entity actually takes when it makes the decision. However, if the link decision is logic based (or shortest path to specific destination for example), the link name returned may be useful to some users in advance of the entity moving there.

Navigation Window Enhancements

We have enhanced the Navigation window with the ability to re-order project items in the project window via drag and drop. Note in the below window, the ModelEntity has been moved below the various models. Moving a model can be done by simply highlighting, dragging and dropping it. A small blue curved "insert here" arrow will appear when moving it to display where it will be located when dropped into a new location.



Note that the various sections of the 'tree' are collapsible as well. When you click on one of the "v" symbols next to a model that has multiple experiments, for example, it will close, as shown below and then contain a ">" next to it instead of a "v".



Note that users can re-arrange the order of *sibling* items (nodes) by dragging them around. That is, you can re-order items within the same parent node:

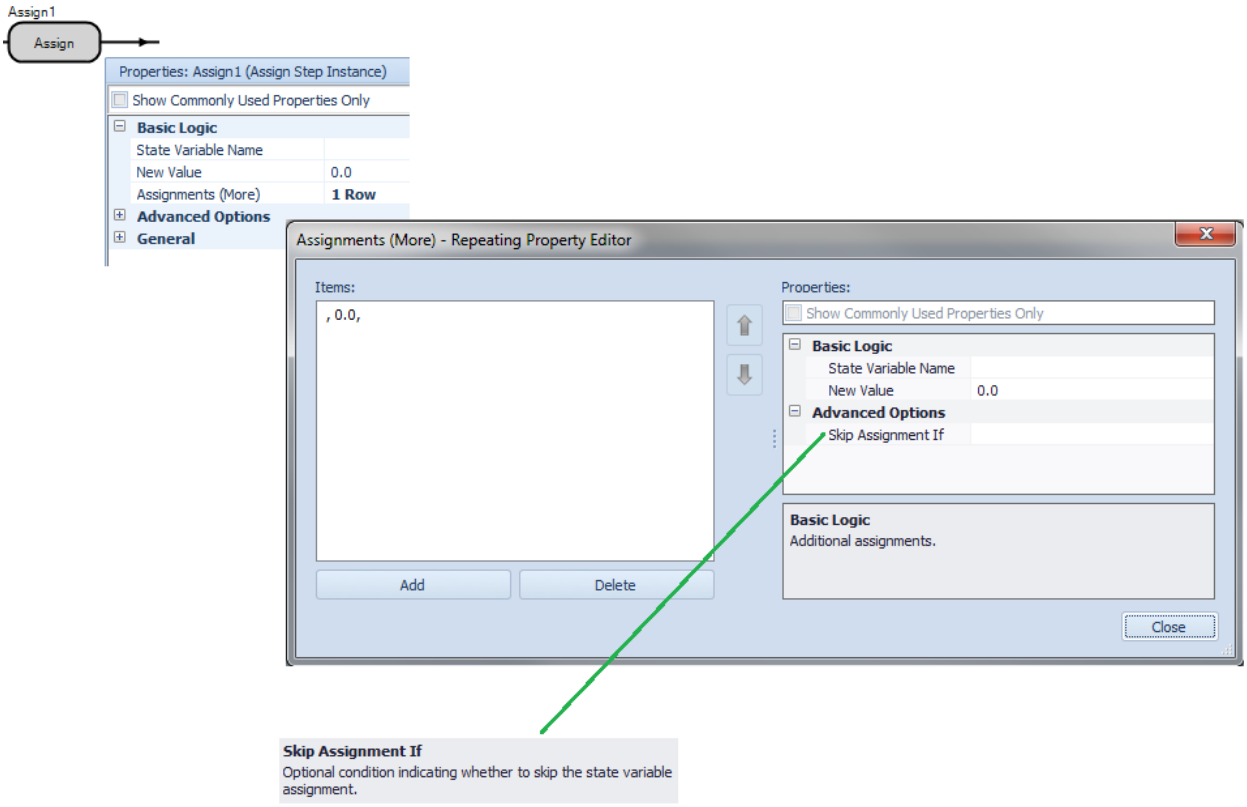
1) Top-level models within the project
2) Experiments within their parent model (actually, the "Experiments" node for that model)
3) Baseline models within their parent model (similarly, the "Baselines" node for that model)
4) Symbols within the "Symbols" node

Simio Release 8 – Sprint 133 – January 7, 2016

In this sprint we have made a number of entity travel related enhancements, including the ability to have entities (ModelEntity, Vehicle and Worker objects) easily move between free space and network based travel. The BasicNode and TransferNode have also been enhanced to include a state assignments option upon entering the node.

Assign Step Enhancement

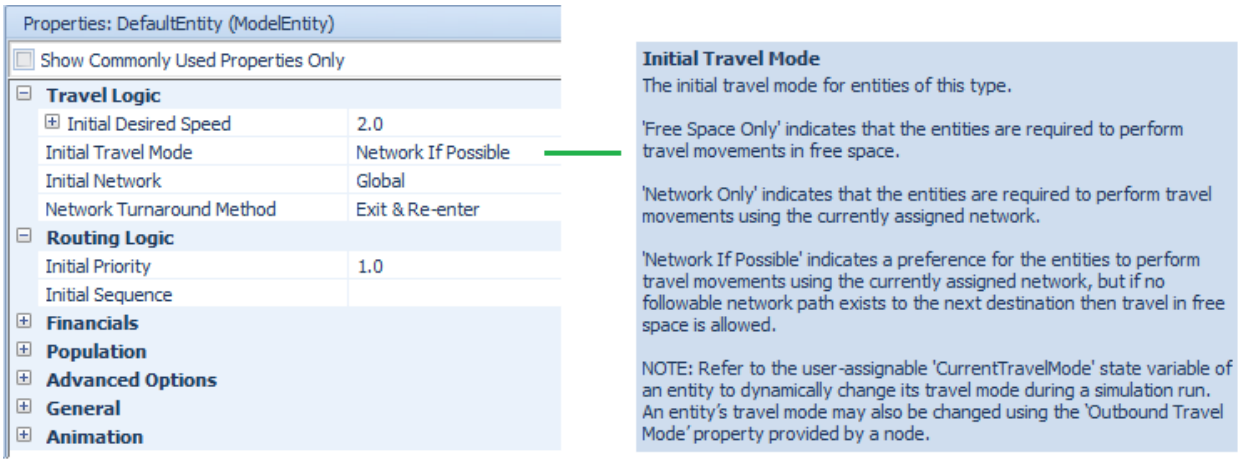
The Assign step has been enhanced to include a *Skip Assignment If* property within the Assignments (More) to allow an assignment in the repeat group to be conditionally skipped. This new feature enabled the node state assignments described below. It will also be useful for conditional type assignments within a single Assign step.



Entity Movement Enhancements

All Entity type objects, including ModelEntity, Worker and Vehicle include a new property within the Properties window, *Initial Travel Mode*. This property can be set to 'Free Space Only' (entity is required to perform travel movements in free space), 'Network Only' (entity is required to perform travel movements using its currently assigned network), or 'Network If Possible' (a preference for the entity to perform travel movements using its currently assigned network, but if no followable network path exists to its next destination then travel in free space is allowed).

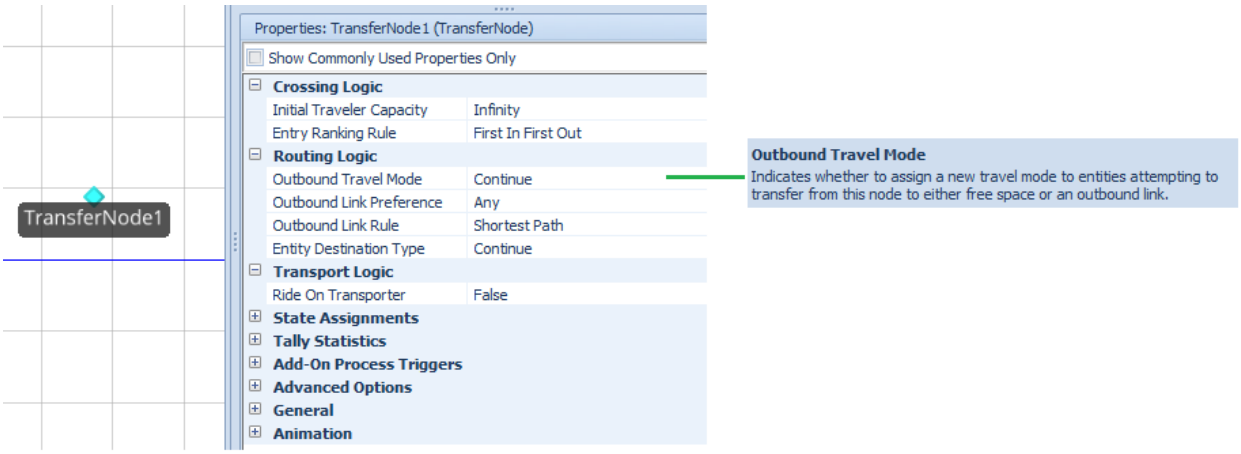
COMPATABILITY NOTE: A model that has extra process logic executing Transfer to 'FreeSpace' steps for an entity that has an assigned network value, rather than relying on the Standard Library object behaviors where assumption is that Entity.CurrentNetwork=='Nothing' to transfer an entity into free space, may have issues with the new *Initial Travel Mode* property set to 'Network Only' (default for existing models updated to sprint 133). Entities traveling between networks and free space should have the *Initial Travel Mode* property changed to 'Network If Possible'.



If the entity then attempts to travel under the wrong conditions, the Transfer step 'Failed' exit point may be exited. See Help on the Transfer step for more details. Additionally, for Worker/Vehicle Pickups/Dropoffs & Resource Move Requests, if a Vehicle/Worker current travel mode is 'Network If Possible', then the Vehicle/Worker will accept any node in the same Facility view as a valid destination for pickups, drop-offs, processing tasks, etc. (because the Vehicle/Worker's travel movements will be able to be performed using either its currently assigned network or in free space).
Related to the above property on the Entity type objects is the addition of a state variable, CurrentTravelMode, that can be assigned during the simulation run to the same options – Free Space Only (0), Network Only (1) or Network If Possible (2). With this, the read-only CurrentNetwork function previously provided for an entity has been changed to be an assignable state. This allows the use of either an Assign or SetNetwork step to dynamically change an entity's currently assigned network.

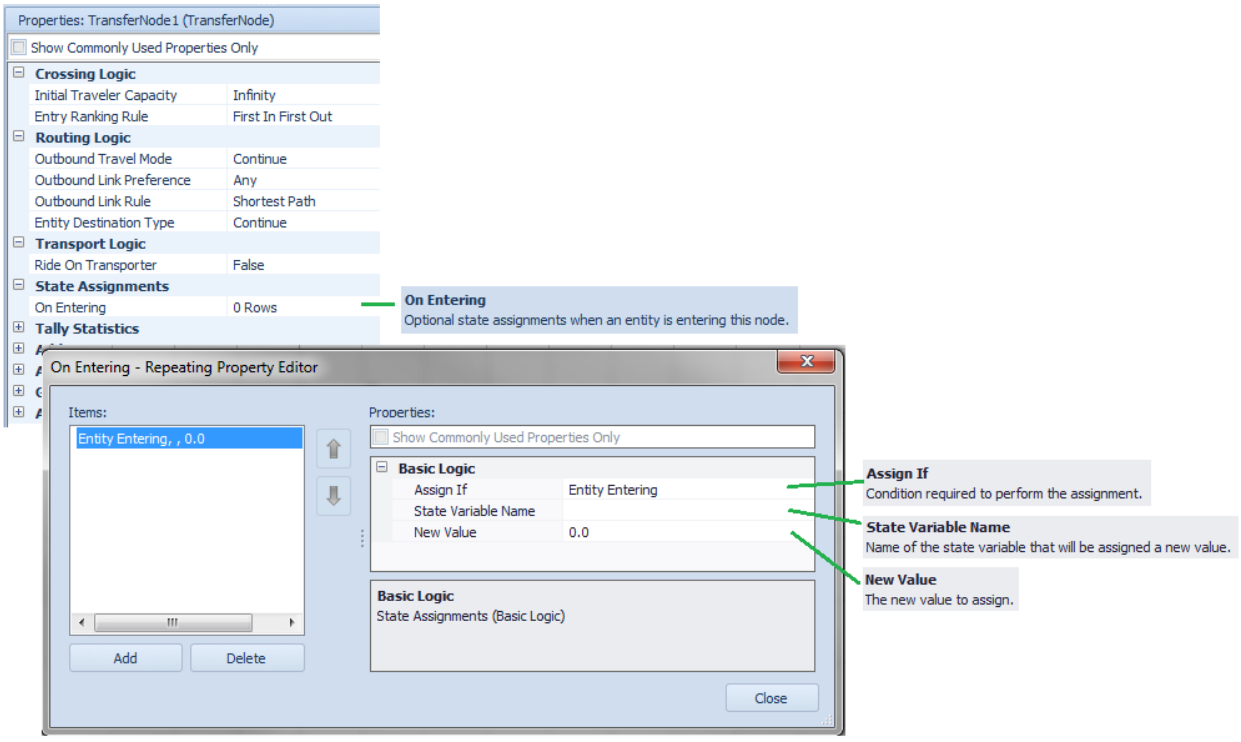
BasicNode and TransferNode Enhancements – Outbound Travel Mode and State Assignments on Entering

In conjunction with the Travel Mode for entities described above, the BasicNode and TransferNode objects now include an *Outbound Travel Mode* property. This property value can be set to 'Continue', which will provide no assignment to the entity's CurrentTravelMode state. The entities will continue using their currently assigned travel modes. The property can be set to 'Free Space Only', 'Network Only' or 'Network If Possible' which will assign a new travel model to the entity.



Within both the BasicNode and TransferNode, we have added State Assignments repeatable properties on entering a node. Because transporters commonly travel through nodes, the state assignments functionality provided by the objects will make it easy for a user to distinguish between entity or transporter state assignments by providing an *Assign If* condition option.

Note that the available assign condition type choices (Custom Condition, Entity Entering, No Condition, Transporter Entering) are identical to the tally condition choices if defining a tally statistic using the node's *Tally Statistics -> On Entering* repeat group.



Simio Release 8 – Sprint 131/132 – December 2, 2015

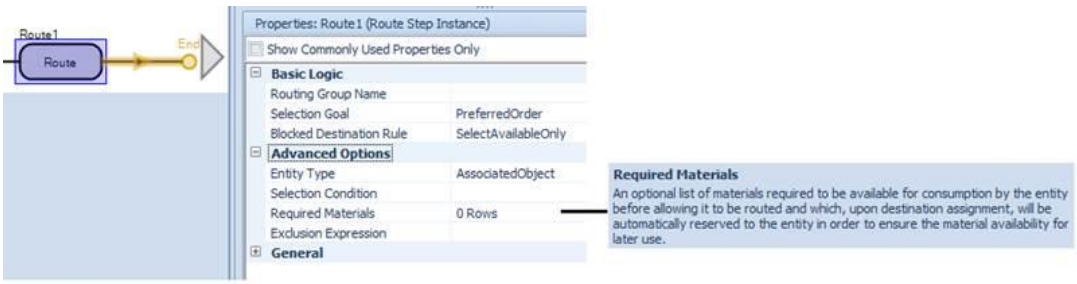
Welcome to Simio version 8 – a compilation of all the features incrementally introduced since version 7 last November. In this sprint, we have enhanced the Route step and the TransferNode object to include reservation of materials before movement, a feature set we have been working on the past few sprints. We have also added some user requested features, such as Independent Probabilistic option for Task Sequences, Secondary Resources release options and Resource Gantt enhancements. And we have added new scheduling documentation and examples.

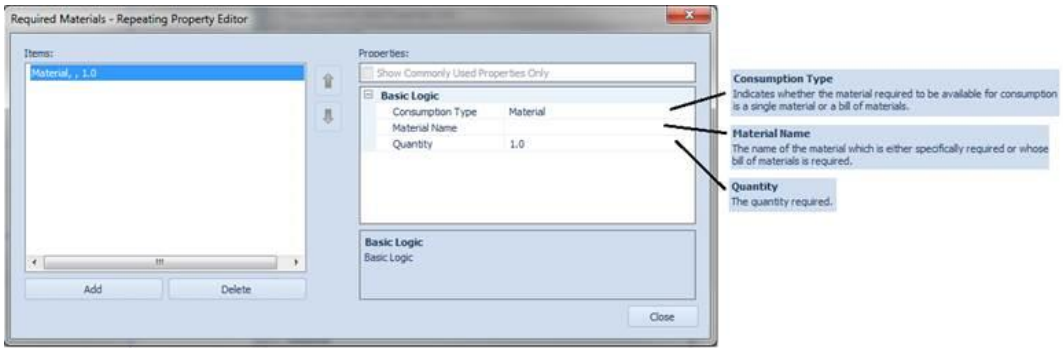
NEW Simio Portal Edition

Simio Portal Edition is now available! Portal Edition brings the power of Simio simulation to the Cloud. With Simio Portal Edition, you can instantly access multiple powerful simulation processors for running simulation experiments and generating plans. Access Simio Portal Edition from an internet connected device using Microsoft's Azure platform, or from Simio's desktop products without having to install any extra hardware or software. Instantly share model results over the web with no software or downloads required. See our Simio Portal page at http://www.simio.com/products/simio-portal.php. For more information on purchasing a license for the Simio Portal Edition or to schedule a demonstration of the product, please contact sales@simio.com.

Route Step Enhancement

The Route step has been enhanced to include a *Required Materials* repeating editor to include multiple materials (single materials and/or bill of materials) and quantities that will be reserved for consumption at a downstream location.



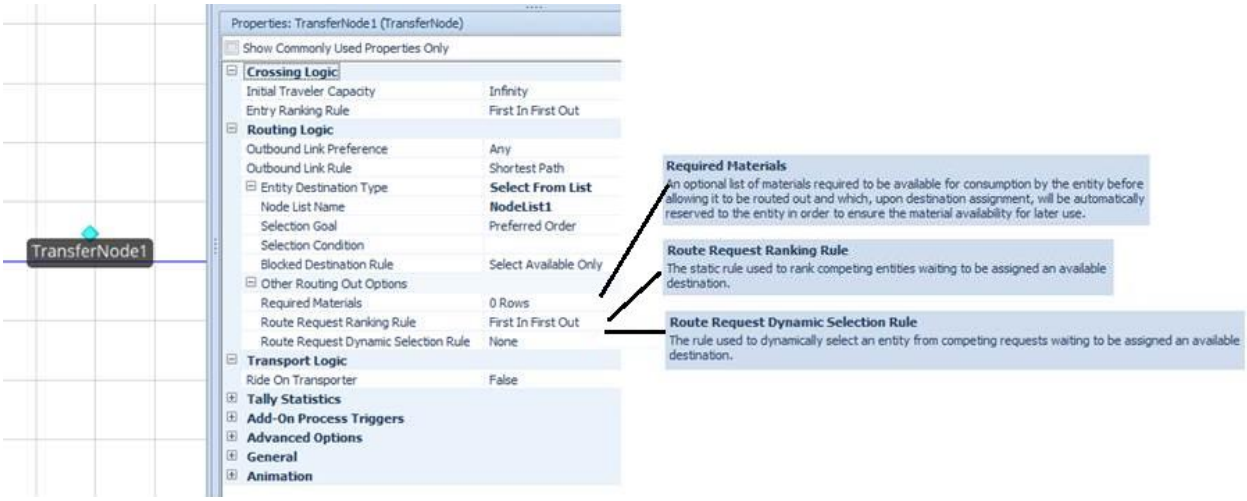


TransferNode Object Enhancements

The above Route step enhancement has been added within the TransferNode object (and thus many other objects containing it) within the Standard Library. The *Required Materials* repeat group properties are available if *Entity Destination Type* is either 'Select From List' or 'Use Custom Routing Group'. The *Consumption Type*, *Material Name* and *Quantity* properties are available for specifying multiple materials required/reserved prior to moving to the destination location.

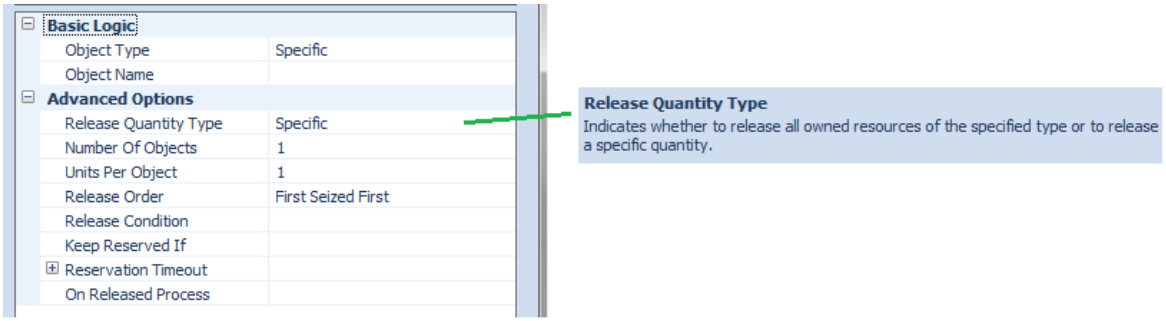
The *Route Request Ranking Rule* property is available if using the 'Select From List' option for *Entity Destination Type*. For the 'Use Custom Routing Group' *Entity Destination Type*, the *Route Request Ranking Rule* property is available on the referenced RoutingGroup element.

The *Route Request Dynamic Selection Rule* property is also available if using the 'Select From List' option for *Entity Destination Type*. See the Dynamic Selection Rules section of help for more information.



Secondary Resource Releases – Release Quantity Type

For the Server, Combiner, Separator, Emptier, and Filler 'Secondary Resources -> Other Resource Releases' features, the user now has an advanced option property allowing the release 'All' owned resources of the specified type or a 'Specific' quantity.



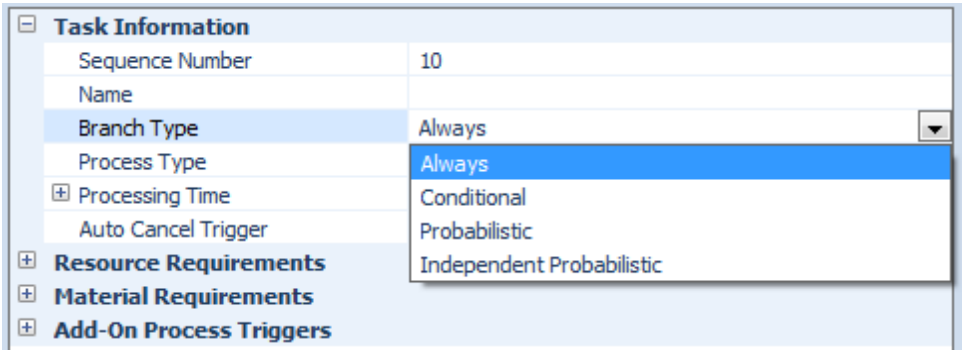
Task Sequences Enhancement – Independent Probabilistic

Within the Task Sequences type of processing in the Server, Combiner and Separator objects, we have enhanced the Branch Type options to include an Independent Probabilistic choice.

When executing a task sequence during a simulation run, the decision to conditionally or probabilistically select a task (branch) is made when all precedence dependencies for the task have been satisfied and it is time to either start or cancel the task. If the *Branch Type* is 'Conditional', then the task is performed if the specified logical condition evaluates to True. Otherwise, the task is cancelled.

The 'Probabilistic' *Branch Type* may be used to identify the task as one of possibly several mutually exclusive alternatives at a probabilistic decision point. The sum of the branch probabilities at a probabilistic decision point cannot be greater than one. In the case of a 'Probabilistic' branch, a single task will be probabilistically selected from the set of candidate tasks (branches) that share the same immediate predecessor dependencies.

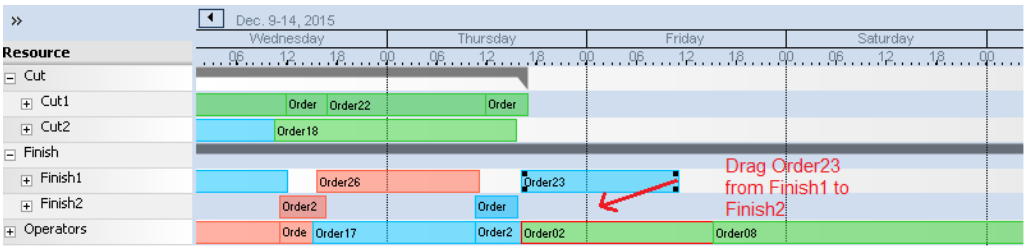
If the *Branch Type* is 'Independent Probabilistic', then the task's specified probability is independent of and thus not affected by other probabilistic branching. For example, if a decision point has two task branches each with an independent probability of 0.5 (50% chance), then that indicates either, both or neither branches might be required to occur. And the probability of both branches being required is (0.5)*(0.5) = 0.25 (25% chance).

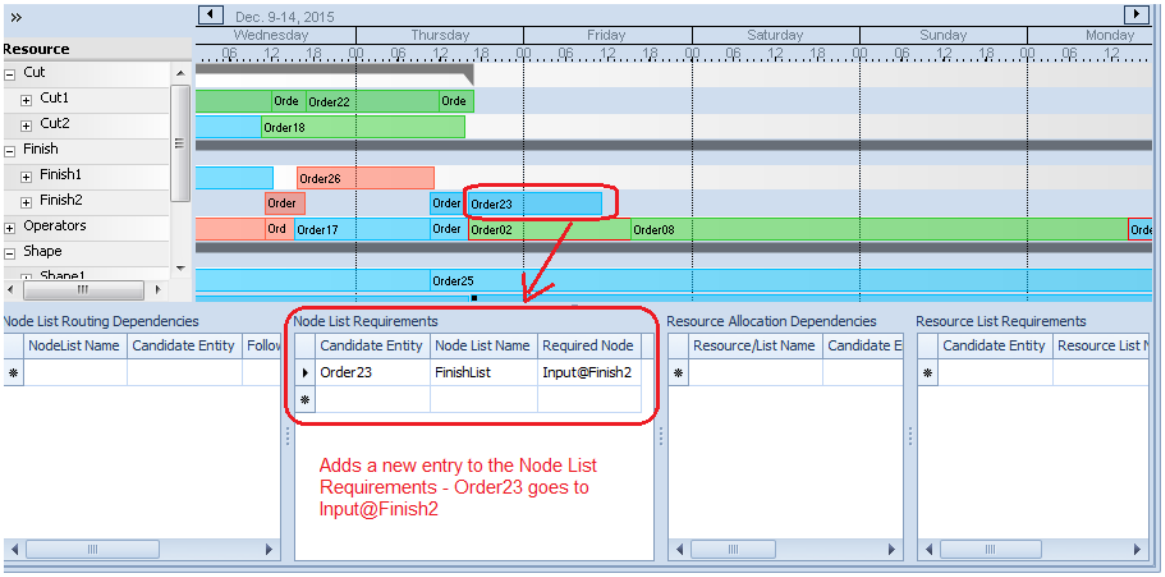


Drag and Drop Node List / Resource Gantt Enhancement (Enterprise Edition)

In previous software versions, after running a plan, requirements that certain entities will seize certain resources from a resource list can be added.

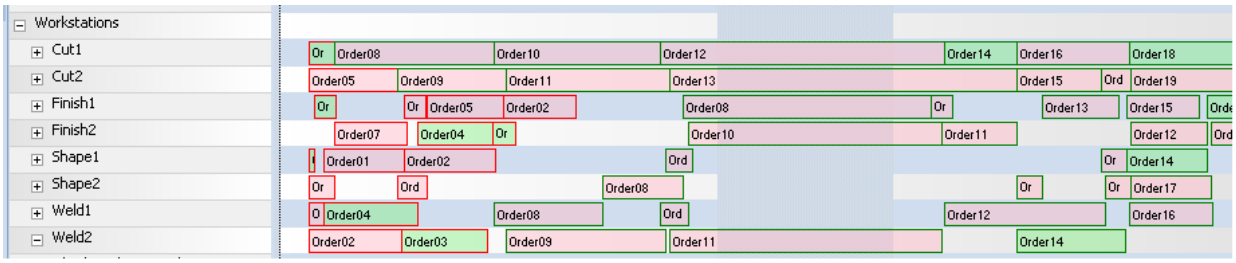
We have enhanced this feature such that a user can now add requirements that certain entities will select certain destination nodes from a node list. Just like the existing resource list requirements, this can be edited directly in the grid, or entities can be moved (drag/drop) from one row to another in the Gantt:

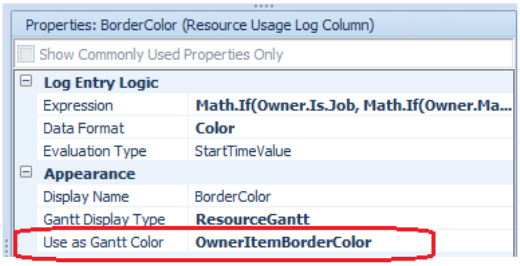




Resource Usage Log / Gantt Enhancements (Enterprise Edition)

Several enhancements have been made to the Gantt charts within Simio Enterprise. In addition to the fill color for the Gantt, users can now add a column to the Resource Usage Log that will allow the border color within the Gantt charts to be customized. In the example shown below, the fill color for the Resource Plan Gantt is pink for priority 1 items, and light green for priority 2 items. The border color is red for 'WIP' and green for 'New' orders. The *Expression* field should evaluate to a particular color (i.e., Color.Red, etc.) when the *Data Format* is 'Color'.





New Documentation for Scheduling

The e-book *Planning and Scheduling with Simio* (available via Support ribbon > Books) has been totally rewritten to reflect recent software changes and better explain the scheduling concepts and overview the scheduling features.

New Scheduling Examples

An existing example was updated and renamed and two additional examples have been created. All three examples (available via Support ribbon > Examples) have extended documentation that is intended to be use along with the *Planning and Scheduling with Simio* book mentioned above.

SchedulingDiscretePartProduction is an updated version of the previous example named RPSExample. It illustrates and explains how to schedule a job-shop operation.

SchedulingBicycleAssembly provides an example of a simple assembly operation.

SchedulingBatchBeverageProduction provides an example of batch processing as is often found in beverage, pharmaceuticals, and fluid-like operations.

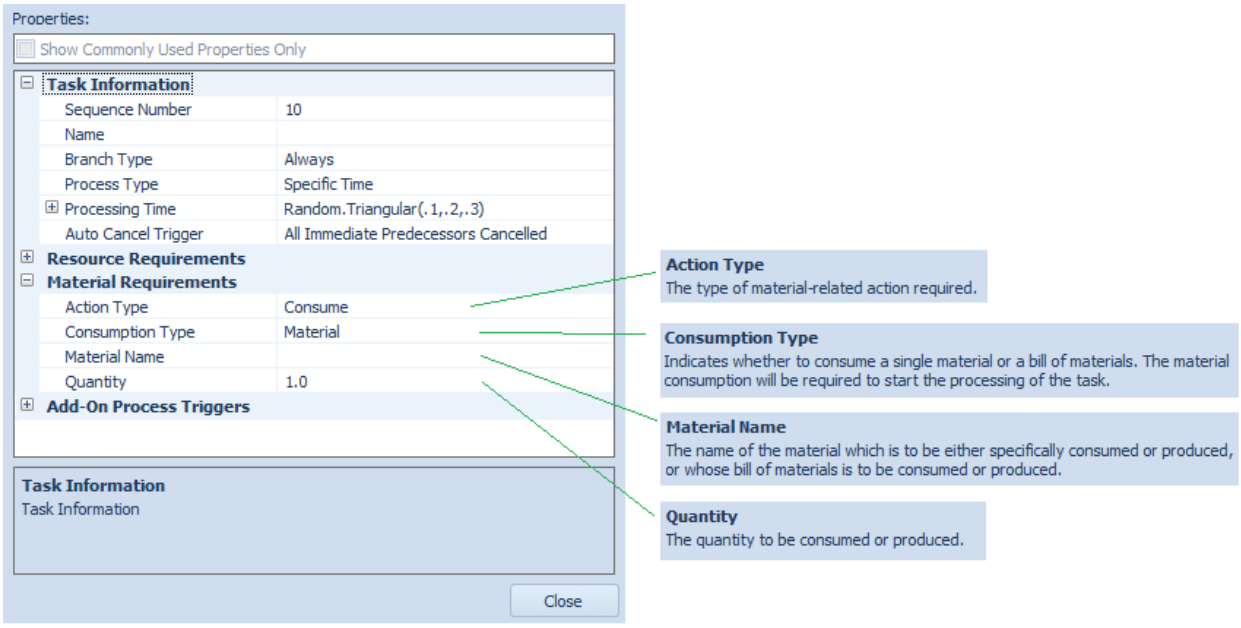
Simio Release 7 – Sprint 130 – November 2, 2015

In this sprint, we have incorporated the materials type changes from Sprint 129 into the Workstation and Server, allowing for multiple materials/bill of materials to be consumed or produced within a given Workstation or a processing task in a Server. The Task Sequences option on processing has been added to Combiner/Separator objects. These features, along with the Reserve steps added last sprint should allow for great flexibility in scheduling operations. We've also added a much requested feature to allow fixed objects to be moved during the simulation run with the RelocateObject step.

Server - Task Sequence - Material Requirements

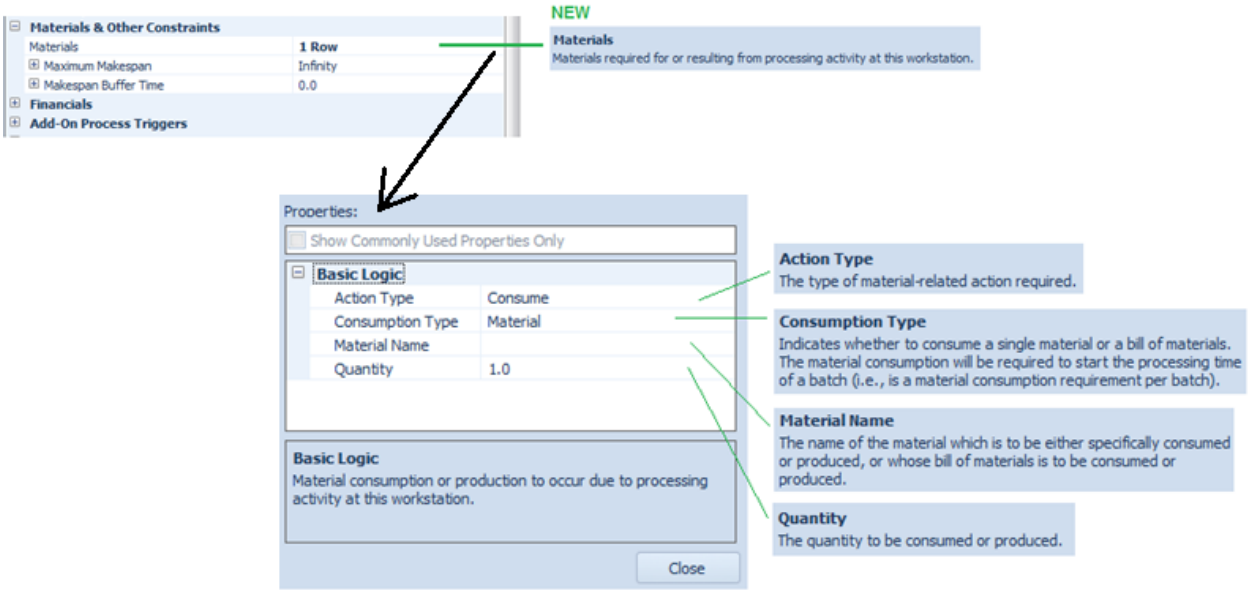
Within the past several months, we've enhanced the Server to include the option of having 'Task Sequence' type processing. With that, the user specifies a number of processing tasks and required resources for each task. We have continued to enhance this feature and have added a Material Requirements section to the processing tasks. With this, the user may specify one or more materials / bill of materials that may be consumed or produced at each particular processing task.

The below example shows the *Action Type* 'Consume' and can alternatively be 'Produce'.



Workstation – Materials Repeating Property Editor

Sprint 129 brought about the repeatable type Produce and Consume steps, allowing the user to specify multiple materials or bill of materials to produce and consume within each step. We have incorporated these features into the Workstation object and now have a *Materials* repeating property editor for specifying materials to consume and/or produce.



Updated SimBits

Based on the above two enhancements, we have updated a number of SimBits to reflect these changes.

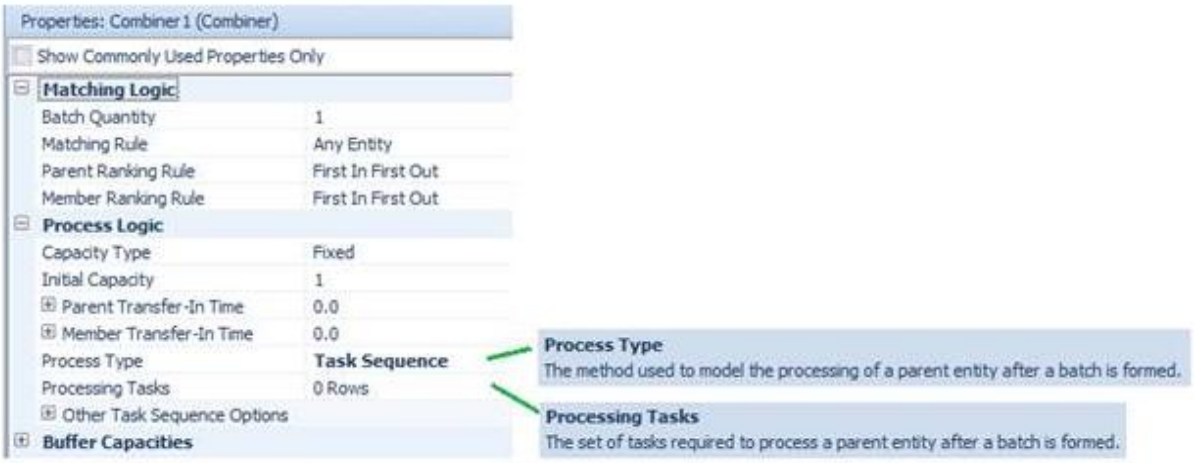
These include the modification of the Workstation/Material related SimBits, including ScheduledMaterialArrivals, ScheduledMaterialArrivalsWithDeviation, WorkstationWithMaterialConsumption and WorkstationWithMaterialConsumptionAndReplentish.

Additionally, we have added Materials to several Task Sequences related SimBits including ServersUsingTaskSequenceWithDataTables_JobShop and ServersUsingTaskSequenceWithDataTables_FlowLine.

Combiner and Separator - New Process Type and Task Sequence Functionality

To keep the Combiner and Separator objects consistent with the Server object, as well as to provide additional functionality within those objects, we have added the several properties for *Process Type* (Specific Time or Task Sequence) and *Processing Tasks* repeating property editor. This allows for multiple processing tasks, along with any resource and material requirements, to be specified within each of these objects.

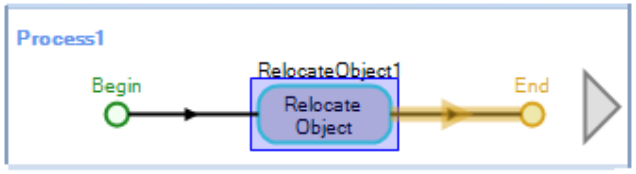
Within the Combiner (below), the processing tasks start after a batch has been formed. Within the Separator, the processing tasks are completed before the batch is split or entity copied.

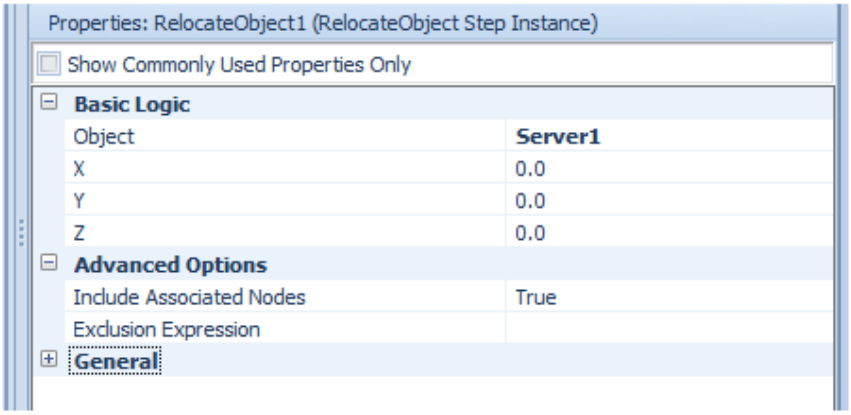


New RelocateObject Step

Within the User Defined steps, we have added a RelocateObject step. This step may be used to instantaneously change the location of an object, including fixed objects and unconnected nodes, at runtime. When relocating or moving a fixed object, such as a Server or Workstation, the *Include Associated Nodes* property set to 'True' will move the object location (and animation) as well as nodes that are attached to the object.

Note that attempting to change the location of a Link or a Node with attached Links will result in a runtime error.



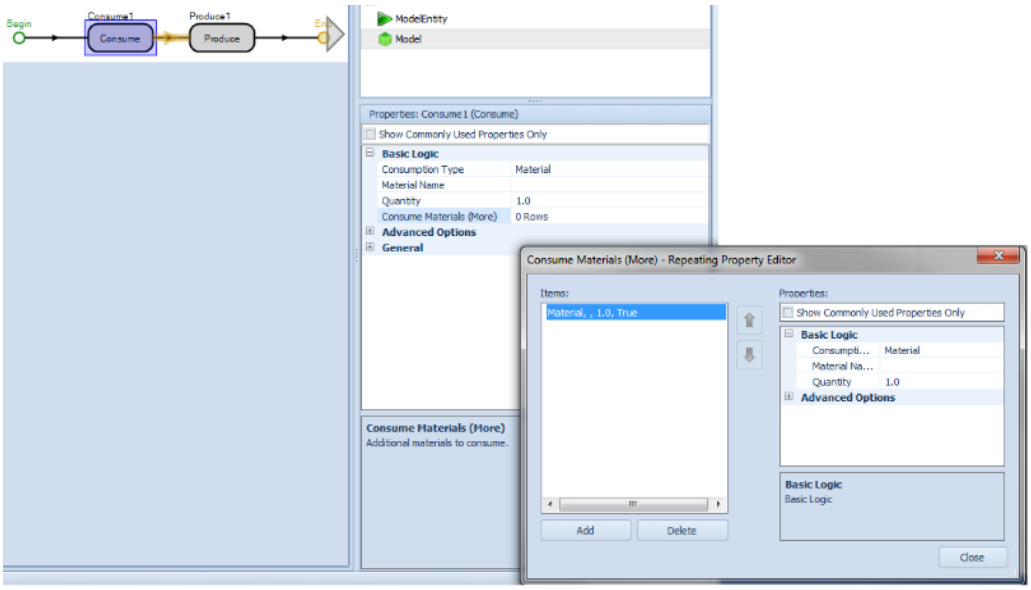


Simio Release 7 – Sprint 129 – October 9, 2015

In this sprint, we have added a number of material related changes, including enhancements to the Reserve, UnReserve, Produce and Consume steps and the addition of several new functions. These material enhancements will continue in the next sprint, with changes to the Workstation object to incorporate the below functionality.

Produce and Consume Step Enhancements

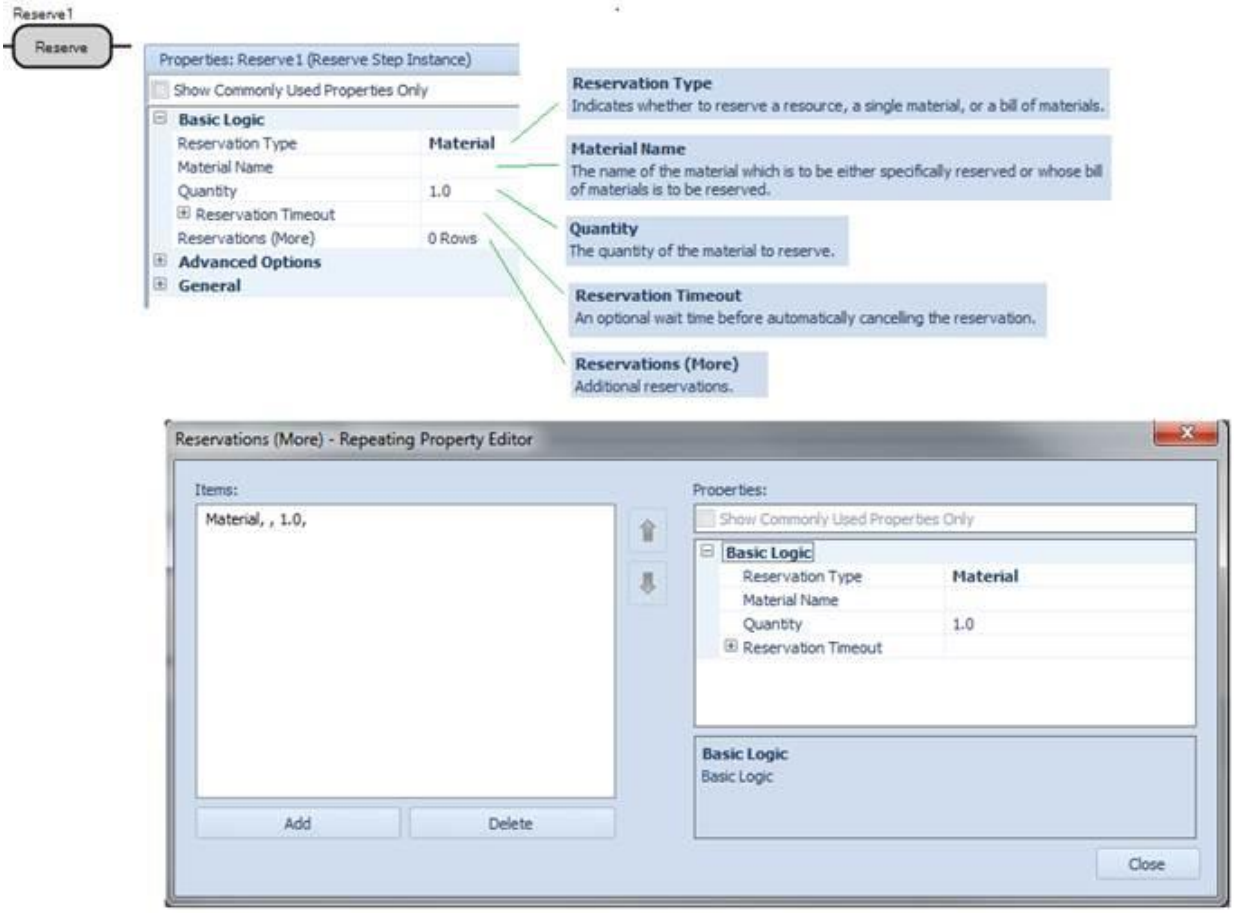
The Produce and Consume steps have both been enhanced to allow the production and consumption of multiple materials / bill of materials within each single step by using the repeating property editor, *Consume Materials (More)*. The below picture shows the Consume step, while the Produce step has similar enhancements.



Reserve & UnReserve Steps Enhancements - Material Reservations

The Reserve and UnReserve steps have both been enhanced to allow materials to be reserved, both simple material and bill of materials. Previously, the Reserve and UnReserve steps were used for only Resources (Resource/Worker/Vehicle reservations).

The *Reservation Type* property has been added, in addition to a repeating property editor, *Reservations (More)*, to have multiple reserve/unreserved requests within the same step. The below picture shows the Reserve step, while the UnReserve step has similar enhancements.



Material Element Functions

The Material element has changed to support the changes for reserving and canceling reserved materials. Two new functions, **QuantityReserved** and **QuantityReservedTo(owner),** have been added to return the total quantity reserved of a given material, as well as the quantity reserved for use by a specified owner.

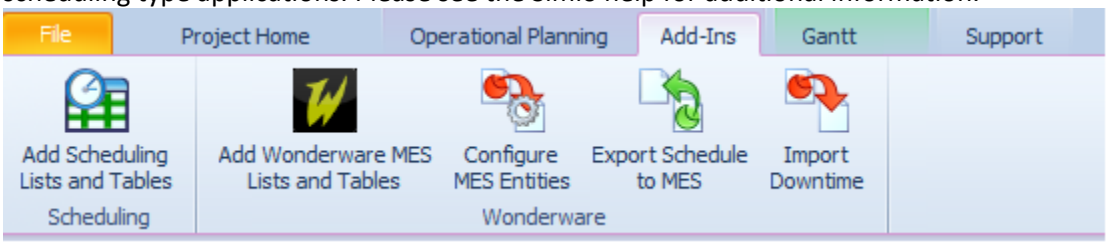
Note that several of the existing state and function names have changed as well. The **QuantityAvailable** state for a material was renamed to **QuantityInStock,** while the **Average / Minimum / MaximumQuantityAvailable** functions are now **Average / Minimum / MaximumQuantityInStock**.

Math.SumOfSamples function

We have added a new math function, Math.SumOfSamples(randomExpression,numberOfSamples) that independently samples a specified random expression for a specified number of times and returns the total sum of the samples. See Simio Help for additional information about this function.

Planning Tab – Add-Ins Ribbon buttons (Enterprise Edition)

Within Simio Enterprise edition, the Planning tab is used to generate plans and evaluate risk for a given model. We have recently added some scheduling type add-ins to the Add-In ribbon for specific scheduling type applications. Please see the Simio help for additional information.

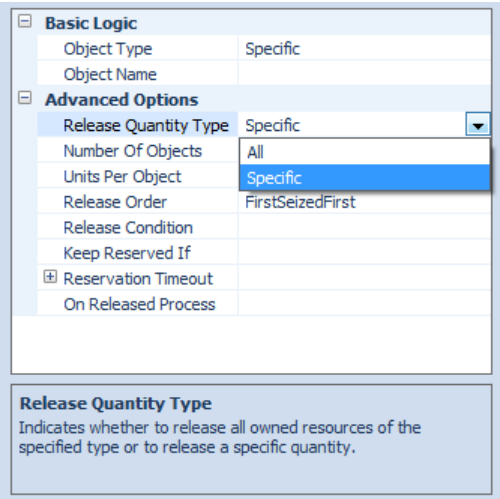


Simio Release 7 – Sprint 128 – September 16, 2015

In this sprint, we have continued to enhance the Task Sequence / Processing Tasks features within the Server to provide the flexibility of specifying multiple resources for any given task. We have also added a number of new SimBits displaying these new Server features as well as other Simio features.

Release Step Enhancement

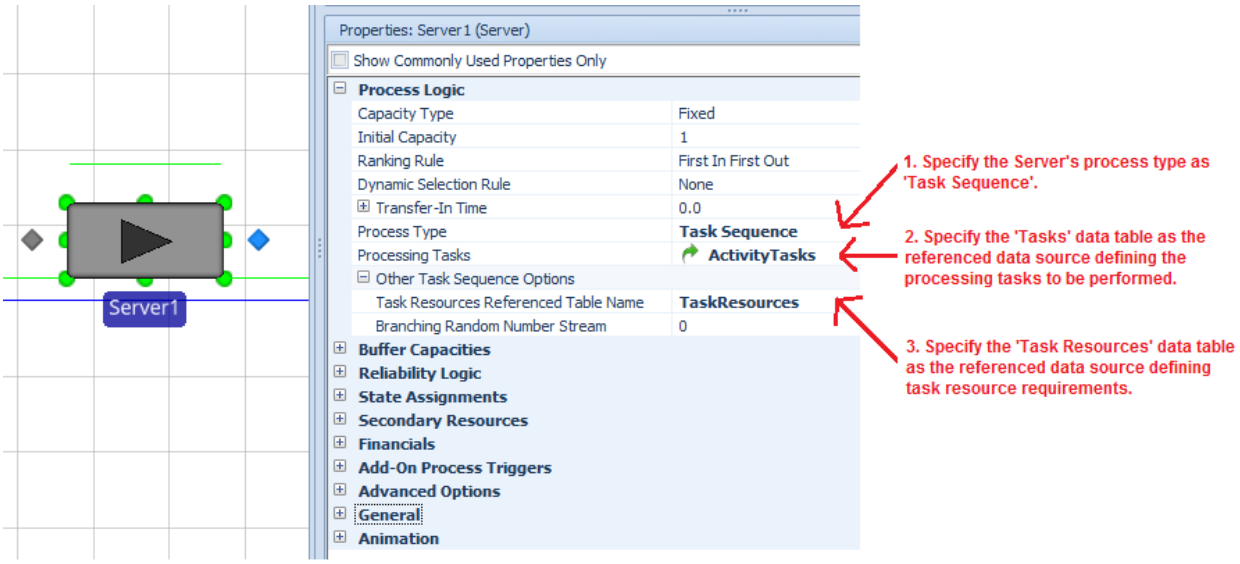
The Release step has been enhanced with a *Release Quantity Type* property that allows the step to either release all owned resources satisfying the selection criteria or release only a specific quantity. This enhancement was added in conjunction with the latest Server changes described below.



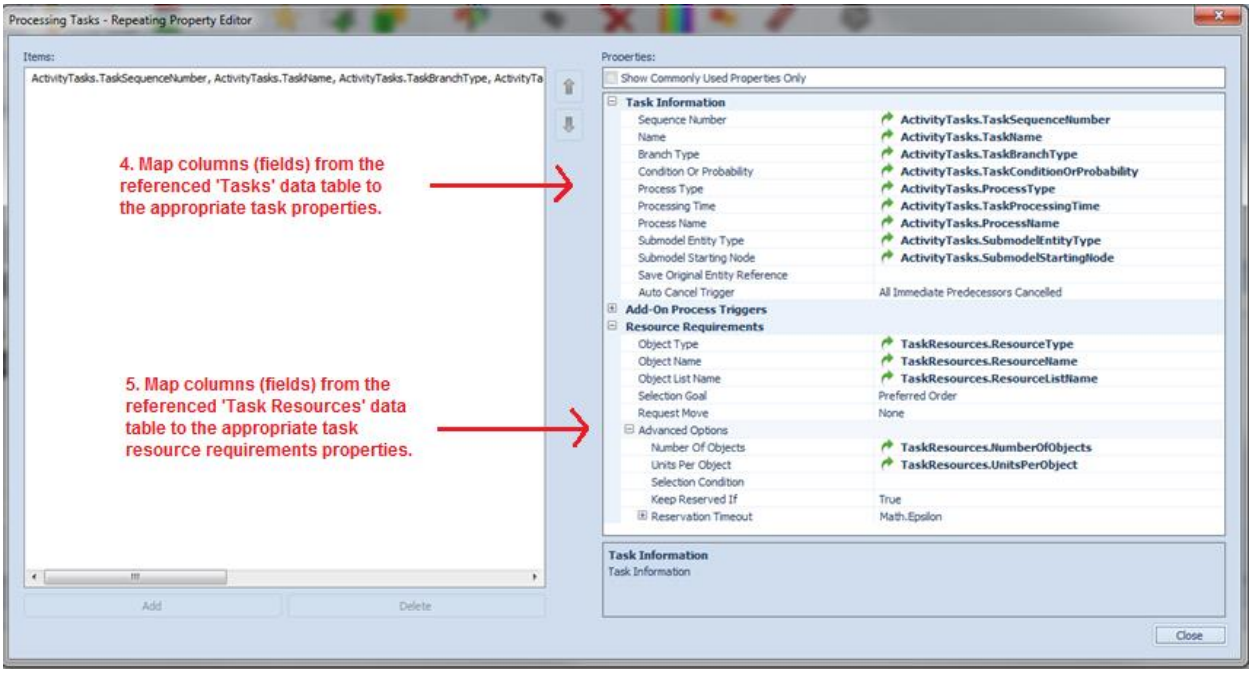
Server Enhancement – Addition of Task Resources Table for Task Sequences

The Server has been enhanced to include the use of **multiple** task resources within a given processing task. In recent sprints, we have enhanced the Server to include the *Process Type* property with 'Task Sequence' type processing (i.e., multiple tasks can be performed within a given Server, where a **single** resource may be required for that particular task).

Several properties have been added to the Server within the Other Task Sequence Options section including *Task Resources Referenced Table Name* and *Branching Random Number Stream*. These are shown below.



Then within the *Processing Tasks* repeat group, a relational table may be referenced within the Resource Requirements section of properties, as shown below.



We have developed two SimBits examples using the relational tables and new Server features for task resources. Please refer to those for additional information.

New SimBits

We have added 5 new SimBits to our library of small examples. The first two of these listed were developed in conjunction with the above Task Resources table approach within the Server, as noted above.

ServersUsingTaskSequenceWithDataTables_FlowLine.spfx – This model demonstrates the use of the task sequence options within a Server, including the use of multiple task resources listed in relational tables. In this example, the same entity type flows through all servers in the system and the tables include server specific data.

ServersUsingTaskSequenceWithDataTables_JobShop.spfx - This model demonstrates the use of the task sequence options within a Server, including the use of multiple task resources listed in relational tables. In this example, the different entity types with varying sequence tables flow through various servers in the system and the tables include server/sequence specific data.

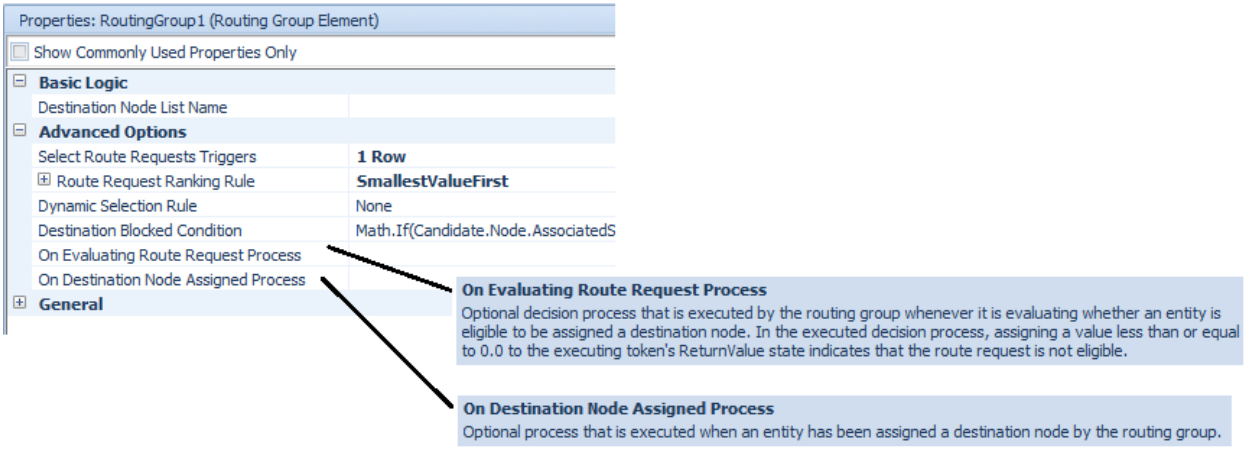
ElectricVehicle.spfx – This model shows an example of a subclassed Vehicle object in conjunction with a level state variable and monitor elements that are used to track 'battery' usage of the vehicle. Several new processes are added to the Standard Library vehicle to evaluate the movement rate of the vehicle and change the battery usage accordingly.

CustomUnbatching.spfx – This model utilizes the Combiner object in conjunction with the UnBatch step to unbatch only certain entities from the group using the Match Condition property on the step.

WorkerUsesWorkSchedule_InterruptWorkingOffShift.spfx – This model demonstrates the use of workers and their schedules including offshift processing. When the workers are used as secondary resources within the Server, the Interrupt step and additional logic is used to halt processing of the entity by both the resource and associated server while the worker is not available.

Routing Group Element Enhancement

The RoutingGroup element has been enhanced to include two new properties that may optionally specify process names to be called during the destination selection process. The first*, On Evaluating Route Request Process*, allows the user to call the process during the destination node decision process. A token ReturnValue of less or equal to 0.0 from this process indicates that a particular entity isn't eligible for the destination node. A second process, *On Destination Node Assigned Process*, will be executed once the destination node is assigned. Both of these options provide greater flexibility within this particular element.


Properties: RoutingGroup1 (Routing Group Element)
Show Commonly Used Properties Only
Basic Logic
Destination Node List Name
Advanced Options
Select Route Requests Triggers
1 Row
Route Request Ranking Rule
SmallestValueFirst
Dynamic Selection Rule
None
Destination Blocked Condition
Math.If(Candidate.Node.AssociatedS
On Evaluating Route Request Process
On Destination Node Assigned Process
General
On Evaluating Route Request Process
Optional decision process that is 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 value less than or equal to 0.0 to the executing token's ReturnValue state indicates that the route request is not eligible.
On Destination Node Assigned Process
Optional process that is executed when an entity has been assigned a destination node by the routing group.

Simio Release 7 – Sprint 124 – June 30, 2015

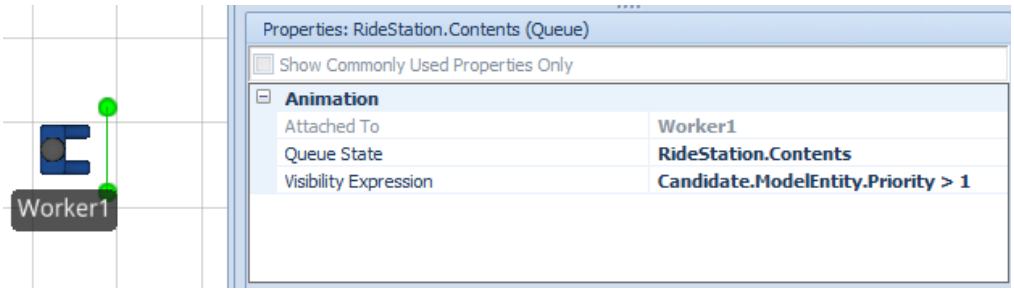
This sprint our mid-year academic-focused release. We've included a number of new animation related features for queues and symbols/symbol filtering, as well as three new SimBits demonstrating some of our latest features (reserving resources for transport and processing tasks). We've also added a few user requested features, such as built-in travel distance tracking, an enhancement on FlowNode output split allocation rule, and random number stream properties to several steps and elements.

Animated Queue Enhancements – Visibility Expression and Match Attached Animation Speed

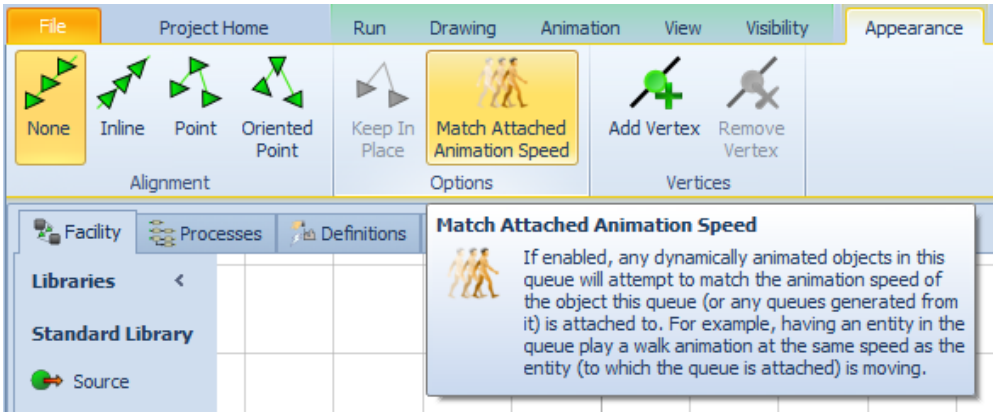
A new property, Visibility Expression, has been added to all animated queues that is evaluated per object in the queue to determine if that object is visible or not. To reference an object in the queue for which the expression is being evaluated, use the "Candidate" keyword.

For example, the "Candidate.ModelEntity.Priority > 1" assumes all objects in the queue will always be of type 'ModelEntity' and will only show those that have a 'Priority' value greater than 1.

This new feature will allow users to place a single queue multiple places graphically while having entities visible based on the visibility expression specified.

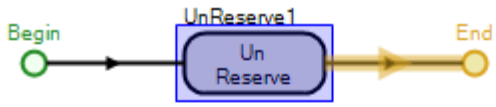


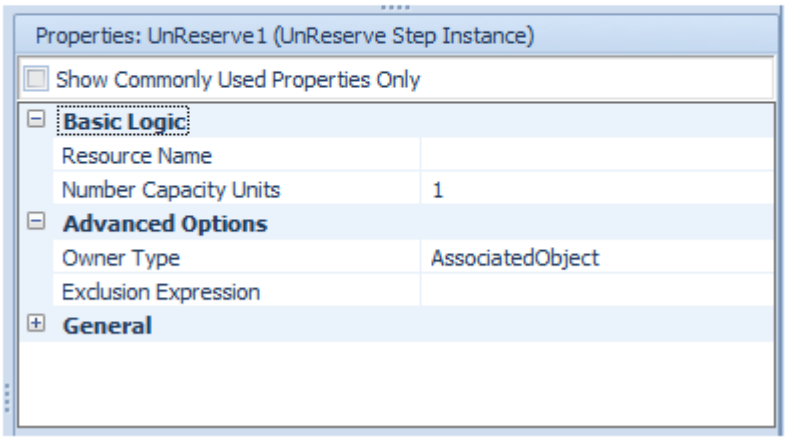
Additionally, for queues that are associated with transporter type objects, such as Vehicle and/or Worker, a new option was added on the Appearance ribbon to 'Match Attached Animation Speed'. If an entity that has been allocated to a Vehicle/Worker is of an 'animated' type (moving legs, etc.), this feature will cause the entity to graphically move along with the Vehicle/Worker instead of gliding (no moving legs).



New UnReserve Step

To balance out last sprint's addition of the Reserve step, a new UnReserve step has been added that can be used in process logic to cancel any reservations for a resource made by a specified owner up to a desired number of reserved capacity units cancelled.





Random Number Stream Enhancement

The following steps and elements now provide a new 'Random Number Stream' advanced option, to allow a user when applicable to optionally specify the random number stream to be used.

1) Timer Element - for use if interval type is 'RateTable' or 'ArrivalTable' (if using no-show probabilities)
2) Route step - for use if destination selection goal is 'Random'
3) StartTasks step - for use if probabilistic branching in the task sequence
4) Seize step - for use if resource selection goal is 'Random'

Distance Traveled Function

A new TotalDistanceTraveled function is available on entities and transporters. This records a cumulative total of the distance that object has travelled. Workers, Vehicles, and other non-destroyable dynamic objects will also have this statistic automatically reported in the output statistics.

SimBit RecordDistanceTraveled

The existing SimBit RecordDistanceTraveled has been updated to illustrate use of TotalDistanceTraveled as well as demonstrate how it can be used for periodic statistics like travel per hour or day.

Flow Library FlowNode - New 'Proportional Based On Link Weights If Possible' output flow split allocation rule

Based on user feedback on the Flow Library, we have added an additional flow split allocation rule to the FlowNode. The FlowNode is included in various Flow Library objects such as FlowSource, Tank, Emptier and ItemToFlowConverter.

When the rule is specified as the new rule ‘Proportional Based On Link Weights If Possible’, then the flow regulator will distribute flow as proportionally as possible to the node’s outbound links using the ratios of the link selection weights, while still adjusting when necessary to maximize total outflow. For example, if there are three outbound links designated to receive 20%, 30%, and 50% of the flow and the outbound link designated for 50% becomes blocked, then this rule will automatically adjust so that the other two outbound links will receive 40% and 60% of the flow until the blockage is cleared.

New SimBits

We have added three new SimBits to our library of small examples to further illustrate some of the Server Task Sequence / Process Tasks features such as Resources.

TaskSequenceAndWorker.spfx - This project includes two models that demonstrate use of the task sequence features to execute a series of tasks from within a Server object. The first model, TaskSequeenceAndWorker_InServer, includes the data for the processing tasks specified within the Server itself. The second model, TaskSequeenceAndWorker_InTable, includes the data for the processing tasks within a table and references to the table within the Server.

KeepingWorkerReserved.spfx - This model demonstrates the concept of ‘reserving’ a worker/vehicle for multiple tasks including transport tasks. These features were added in the past two sprints. In this model, a worker is required to move entities from the Source to a Server. The same worker is then required for processing tasks at the Server and then for movement to the next Server location. The ‘Keep Reserved If’ properties are used after transport and processing to continue to use the same worker for a given entity.

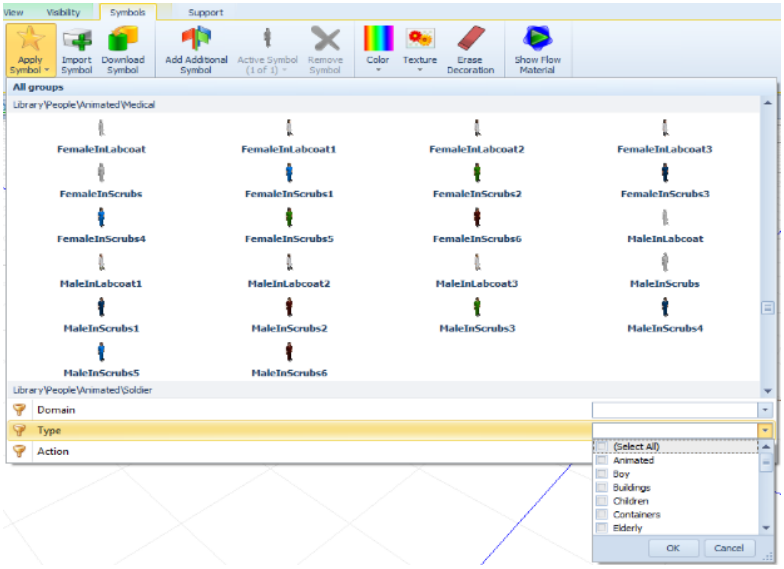
AnimatingQueuesWithVehiclesAndWorkers.spfx – This model demonstrates two of the latest animation features for queues described above, including the Visibility Expression property and the Match Attached Animation Speed option. One portion of the model shows different priority entities in different queue locations on a vehicle, while the other portion of the model shows an animated entity walking with an animated worker.

New Symbols

In addition to the over 100 new static symbols added in the last sprint, we have added Sit, Carry, and Push actions to our existing animated male symbols and we have added 22 new animated people, including various male/female symbols in scrubs and lab coats. While these actions and symbols are particularly useful in healthcare applications, we think they will be useful in many other domains as well.

Symbol Filtering

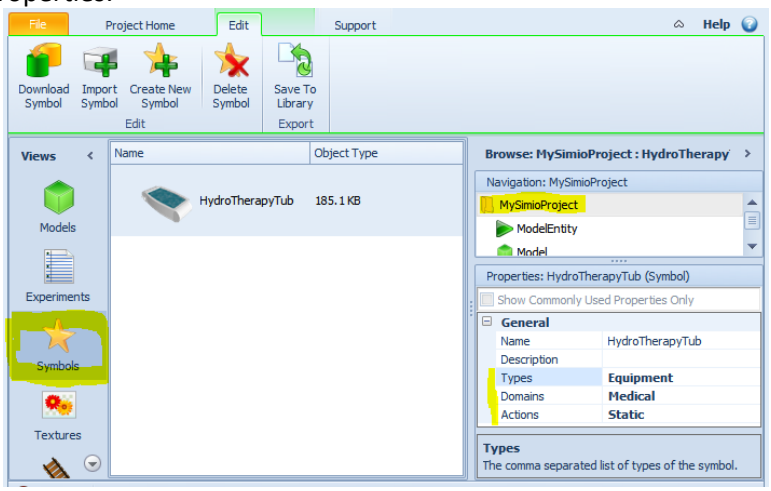
To help you select from this increased number of symbols, we have added filtering. Within the Symbol ribbon, when applying a symbol for an object, new filtering options will help you quickly get to the symbols that interest you. This includes filtering by Domain, Type and Action. **Domain** filtering currently includes Manufacturing, Medical and Restaurant symbols. **Type** filtering includes various types of symbols such as buildings, containers, equipment, furniture, people, and trucks (see screen shot below). **Action** filtering will distinguish between animated and static symbols.



Symbol Creation Features

The ability to create or add your own symbols to the Simio library has been significantly enhanced. We now support the FBX 3D object format and we have significantly enhanced the import capabilities and options to import DAE formatted objects. This better supports the use of external tools like Autodesk Character Creator and Mixamo.

You can also categorize your symbols for use with the filter described above by specifying the data in the Project > Symbols properties:



Simio Release 7 – Sprint 123 – June 8, 2015

This sprint has tied together many of the application specific features with seizing and releasing resources and transporters that we have been working on in the past several sprints. We now have the capability to have resource (including Resource/Worker/Vehicle) objects that can be 'reserved' when they are released so that the entity may either immediately or eventually use the same resource for additional tasks.

Transporter Class Object change including Worker / Vehicle – 'Reserved'

In Sprint 120, we added the capability to the Release step to allow users to keep a resource reserved under particular conditions using the *Keep Reserved If* and *Reservation Timeout* properties. In that sprint, we additionally supported the 'Reserved' capability with Resources. We have now extended this functionality to include the Base Transporter Class, which includes Workers and Vehicles. If a worker or vehicle is 'Reserved' as a Resource, Simio will automatically reject any ride pickup request without reservation.

Ride Step – Enhancements

Reserved Evaluation - When executing a Ride step, if the rider entity attempts to select a transporter from a group of candidates (e.g., from a list or from a population of some specific transporter type), the preference will be to select a transporter resource that the entity has 'reserved'. For example, if an entity is attempting to seize a transporter from a list that contains candidates 'Worker1', 'Worker2', and 'Worker3', and the entity has an active resource reservation for 'Worker2', then the entity will have a preference to select 'Worker2' irrespective of the Ride step's specified reservation method and selection goal.

Token Wait Action - The Ride step provides a new *Token Wait Action* property in its advanced options. This property will give a modeler control over whether the token executing the Ride step immediately continues to the next process step (the only behavior currently supported), or whether the token waits until the rider entity starts or has finished transferring onto a selected transporter.

New OnCapacityReservationCancelled Process

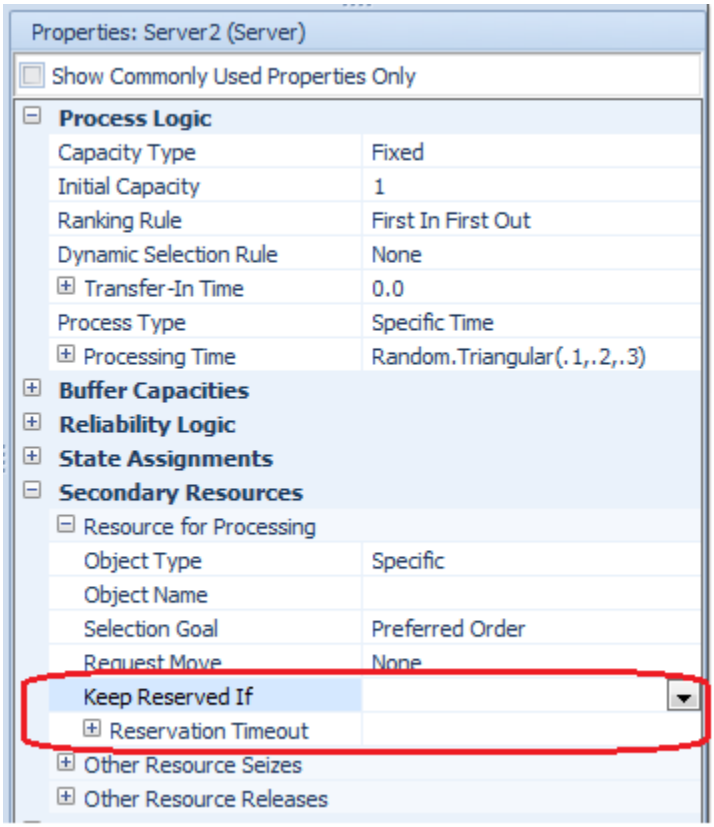
A new *OnCapacityReservationCancelled* interface process is now automatically provided for any resource-enabled object. This process will be immediately executed by the engine whenever a capacity reservation for the resource has been cancelled (e.g., due to a reservation timeout expiring or because an entity owning a reservation was destroyed). Other standard interface processes already provided for any resource object include *OnCapacityAllocated*, *OnCapacityChanged*, and *OnCapacityReleased*.

Currently, within the Resource, Server, Combiner, Separator and Workstation, this process is available when sub-classing or copying an object, but has no specific logic within the process for that object, as the equivalent of an Allocate step is done automatically by the engine for any resource object. Within the Worker and Vehicle objects, this process is used within the object when a reservation is cancelled to allocate the worker/vehicle to seize requests (via Allocate step) and then to transportation requests (via PlanVisit step).

Standard Library 'Secondary Resource' Feature Enhancements - Reserved

For all processor-oriented objects in the Standard & Flow Libraries that support a 'Secondary Resources' property category (e.g., Server, Combiner, Separator, Filler, and Emptier), the option to 'Reserve' a secondary resource when releasing it has been added as an available feature. The *Keep Reserved If* and *Reservation Timeout* properties are available for the Resource for Processing, as shown below, as well as any Other Resource Releases available for the object.

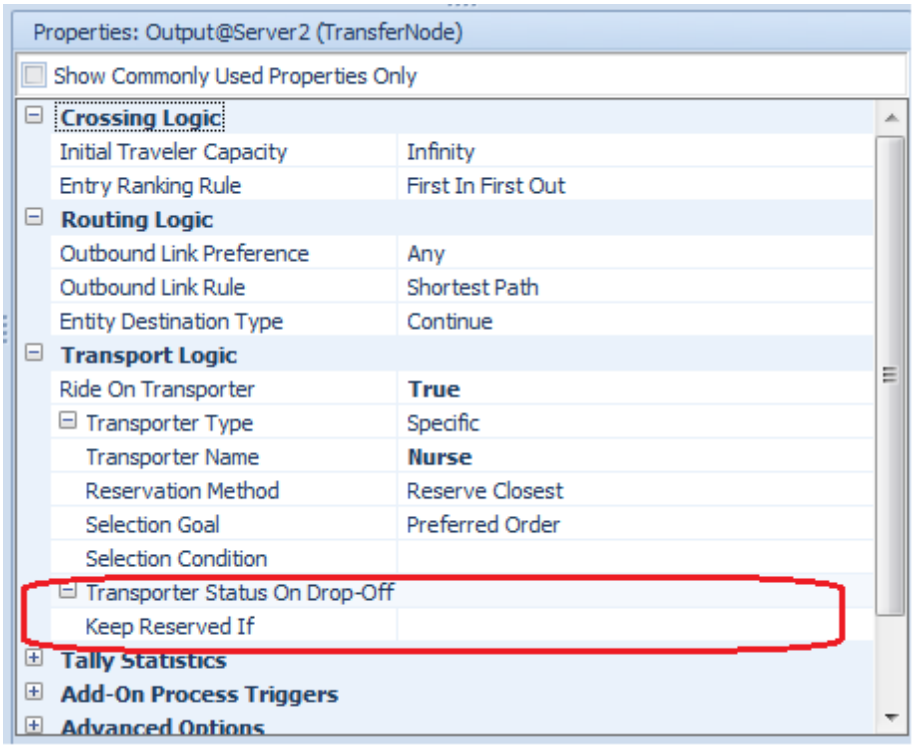
These 'Reserved' type features were originally added to the Release step in Sprint 120 and are now available within the various objects in the Standard and Flow Libraries.



Standard Library TransferNode Enhancement - Reserved

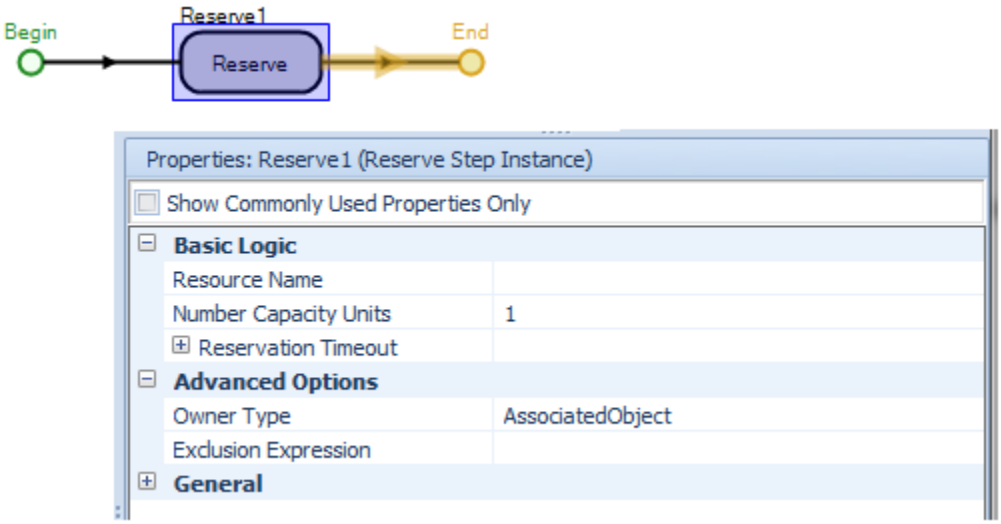
The Standard Library TransferNode has also been enhanced with the 'Reserved' capability through the *Keep Reserved If* property. This will allow a user to specify whether an entity, if riding on a transporter from the node, will keep the used transporter reserved on drop-off. Other entities will be unable to use a reserved transporter for processing tasks or transportation unless the reservation is cancelled.

With this feature incorporated into the TransferNode object, it then is automatically included in all objects with TransferNodes, including Standard Library objects such as Source, Server, Combiner, Separator, Workstation, as well as Flow Library objects such as Filler, Emptier and FlowToItemConverter.



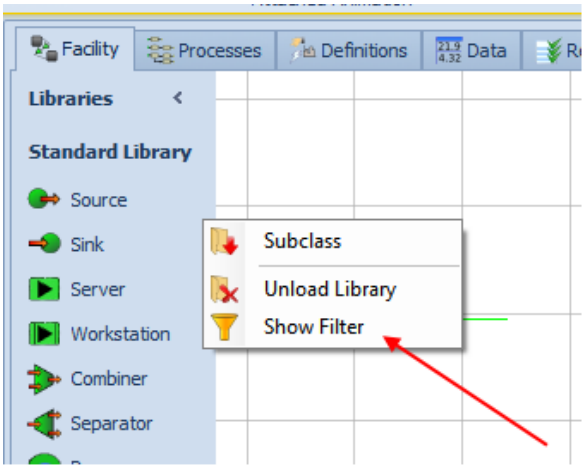
New Reserve Step

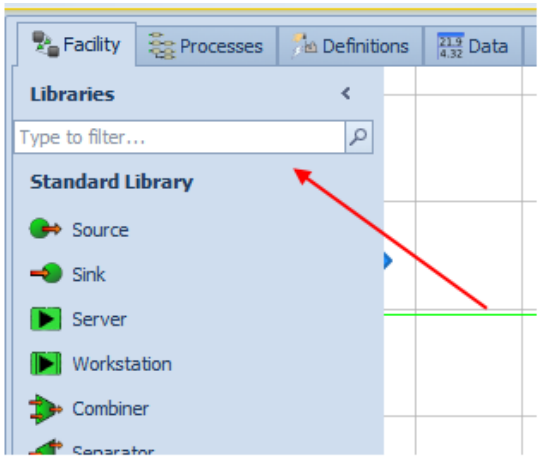
A new Reserve step has been added that will allow a modeler in process logic to arbitrarily reserve a resource's capacity for possible later use. Reserved resource capacity is capacity that, although unallocated, has been set aside for a specific future owner thus preventing its use by anyone else. This can be used in conjunction with the other 'reserved' type features listed above. This step was also required for implementation of the TransferNode enhancement listed below.



Library Objects and Step Filtering

Simio now allows the ability to filter a models list of libraries for specified objects based on the filtering values. To enable this feature, right click on the library panel in the Facility window and select Show Filter. This will display a dialog on the top of the library pane allowing you to enter a value to filter upon.



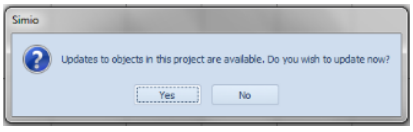


The filter value will apply to all libraries that have been added to that particular model. The Show Filter value applies to each individual model in a project. When a project it saved, the Show Filter value is saved with the project.

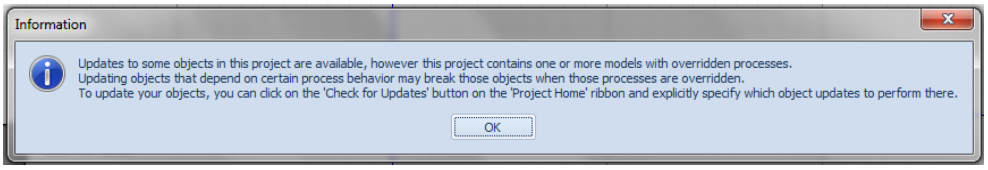
The filtering option is also available in the Processes window. The filter values apply to all buttons within the Processes window. Unlike in the Facility window, there is no option to toggle the filtering within the Processes window.

Object Updating Enhancement

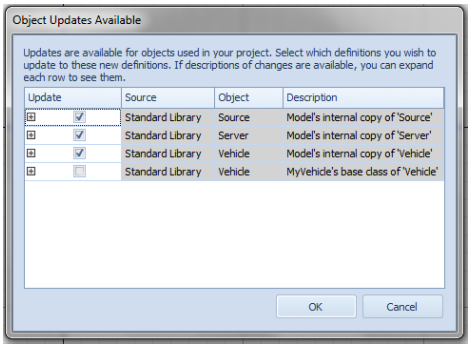
In the past, when Simio makes significant library object updates, there was an update dialog shown with the various objects specified, asking to update (when opening an existing model within a new sprint, for example). Now, if you have no objects with process overrides in your project, you will instead get this dialog:



But if you have some objects with process overrides, will get this instead:



Then when you click the button **Check for Updates**, that dialog will now have objects with overridden processes unchecked by default:



New Symbols

We have added over 100 new “low-poly” symbols (e.g. symbols that are small and fast, but still look good) to the built-in symbol library. These are primarily in the areas of Equipment, Medical Equipment, and Furniture.

Simio Release 7 – Sprint 122 – May 20, 2015

We are continuing our development efforts towards the new Simio Portal Edition, we should be publishing information about Beta software shortly. This sprint we've included a new option on the Release and are working on other application specific features with seizing and releasing resources and transporters. Additionally, a new SimBit displays the use of some resource allocation features that were added to the Server's task sequence / process task functionality.

Release Step – New Advanced Option - Seized Resources Filter

The Release step has been enhanced to include a *Seized Resource Filter* option under the Advanced Options section of properties. This is a filter that will apply to the owner object's list of current seized resources before attempting the resource releases. The options include 'None' and 'TokenSeizedResourcesOnly'. The latter filters the seized resources of the owner object's list to only include resource seizes that occurred specifically due to Seize step execution(s) by the same token now at the Release step.

This functionality was also added to the Server's PerformProcessingTask process used when the Server *Process Type* is 'Task Sequence'.

New SimBit

We have added a new SimBit to our library of small examples to further illustrate some of the Server Task Sequence / Process Tasks features such as Resources.

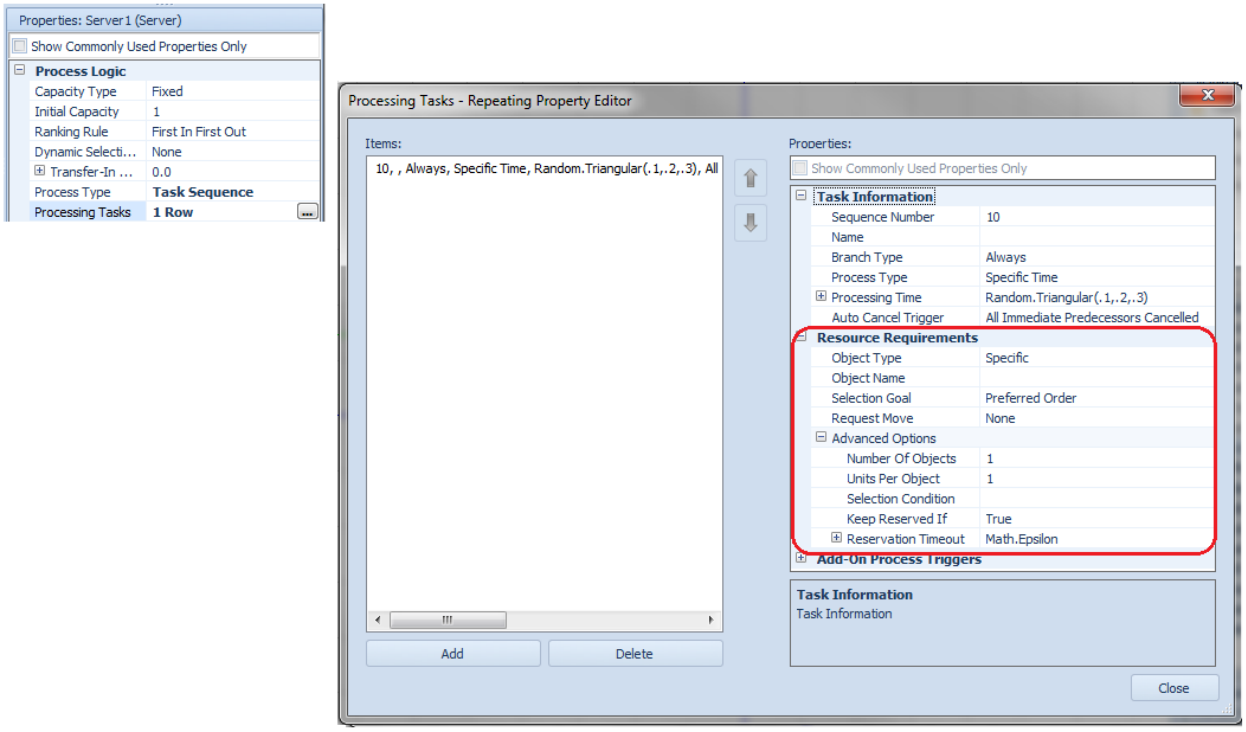
ServerUsingTaskSequenceUsingWorkers.spfx - This model demonstrates the Server object use of the task sequence features to execute a series of tasks from within a Server object. In addition, Worker objects are required for use within each processing task. The use of token functions and resource requirements, both features that were added to the previous sprint, are shown in this new SimBit.

Simio Release 7 – Sprint 121 – April 29, 2015

Our development efforts towards the new Simio Portal Edition are continuing. We have now added Resource functionality to the Task Sequence option within the Server as well as related token-based functions. Additionally, you'll see that we've continued to enhance our Enterprise and Scheduling editions based on feedback from users.

Server Task Sequence (Process Type) Enhancement – Resource Requirement properties

The Server within the Standard Library has been enhanced to include a section of Resource Requirement properties within the *Processing Tasks* repeating properties when using *Process Type* of 'Task Sequence'. This allows for resources (both single or from a list) to be utilized for the various processing tasks. The resource(s) may also then be 'reserved' for future processing tasks.



Note that the default values for the *Keep Reserved If* and *Reservation Timeout* properties above are set to 'True' and 'Math.Epsilon' respectively. If these property values are left unchanged by the user, then by default once the task is finished the entity will always momentarily keep reserved the resource(s) that were seized to perform the task, just in case there happens to be an immediate next task whose specified resource requirements will prefer to re-seize (continue using) the same resource(s).

If there is no immediate re-seize then the resource reservation(s) will be automatically cancelled at the end of the same time step as the epsilon reservation timeout period expires. This default behavior is

intended to make it easy for the user to model the same task resource(s) being used for multiple tasks that are performed consecutively.

New Token Related Functions

TaskInfo.Name – This function returns the name of the task assigned to the token.

TaskInfo.SeizedResources.NumberItems – This function returns the number of task-specific resources currently seized by the task's associated object.

TaskInfo.SeizedResources.FirstItem – This function returns a reference to the first resource in the list of task-specific resources currently seized by the task's associated object.

TaskInfo.SeizedResources.LastItem – This function returns a reference to the last resource in the list of task-specific resources currently seized by the task's associated object.

TaskInfo.SeizedResources.IndexOfItem(resource) – This function returns the one-based index of the first occurrence of a specified resource in the list of task-specific resources currently seized by the task's associated object. If the resource is not found then the value 0 is returned.

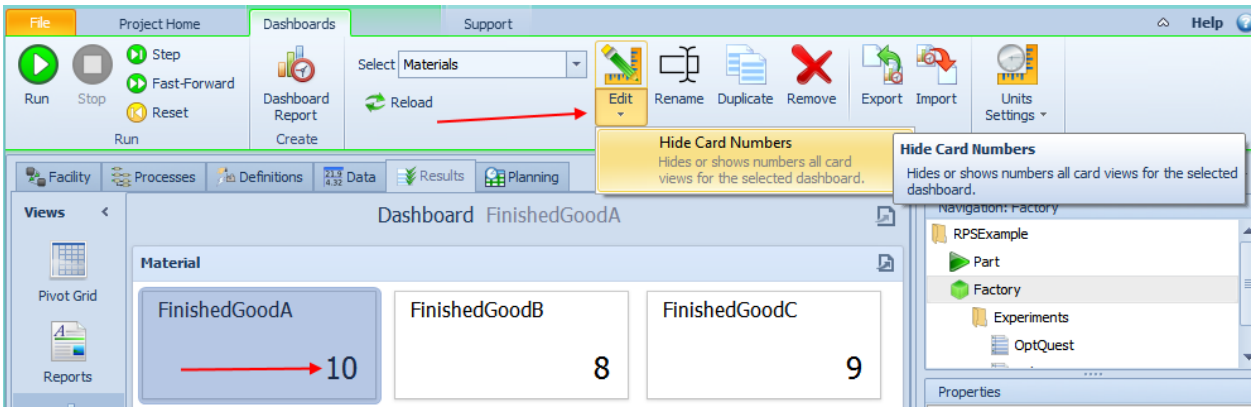
TaskInfo.SeizedResources.ItemAtIndex(index) – This function returns a reference to the resource at a specified index position in the list of task-specific resources currently seized by the task's associated object.

TaskInfo.SeizedResources.Contains(resource) – This function returns True (1) if the list of task-specific resources currently seized by the task's associated object includes the specified resource. Otherwise, the value False (0) is returned.

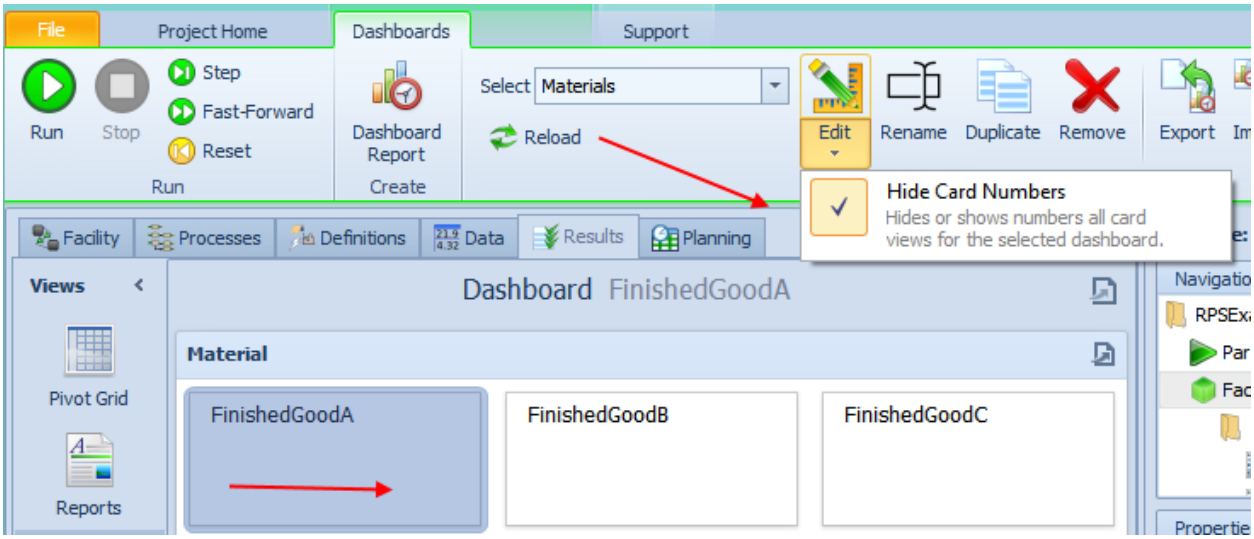
TaskInfo.SeizedResources.CapacityOwnedOf(resource) – This function returns the total number of capacity units of a specified resource, specific to the token's assigned task, that are currently seized by the task's associated object.

Dashboard Reports (Enterprise Edition) – Ability to Hide a Card Data Item's Actual Value from Display

The ability to hide a Dashboard Card's Actual value from the display can be done via the Edit button on the Dashboards ribbon.



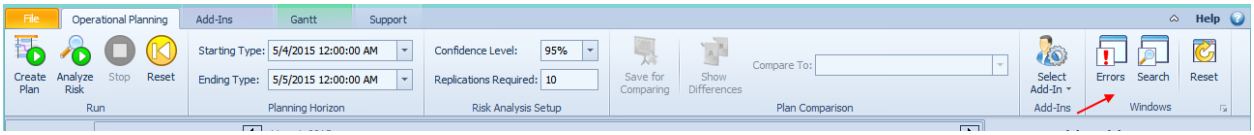
If this toggle is checked, the numbers will not be displayed on the cards as shown below.



Additional Functionality with Scheduler Edition

A Scheduler Edition user now has the ability to view model errors, search via the Search Window and Reset docking windows to their default layout windows. These are features available in all other Simio Editions.

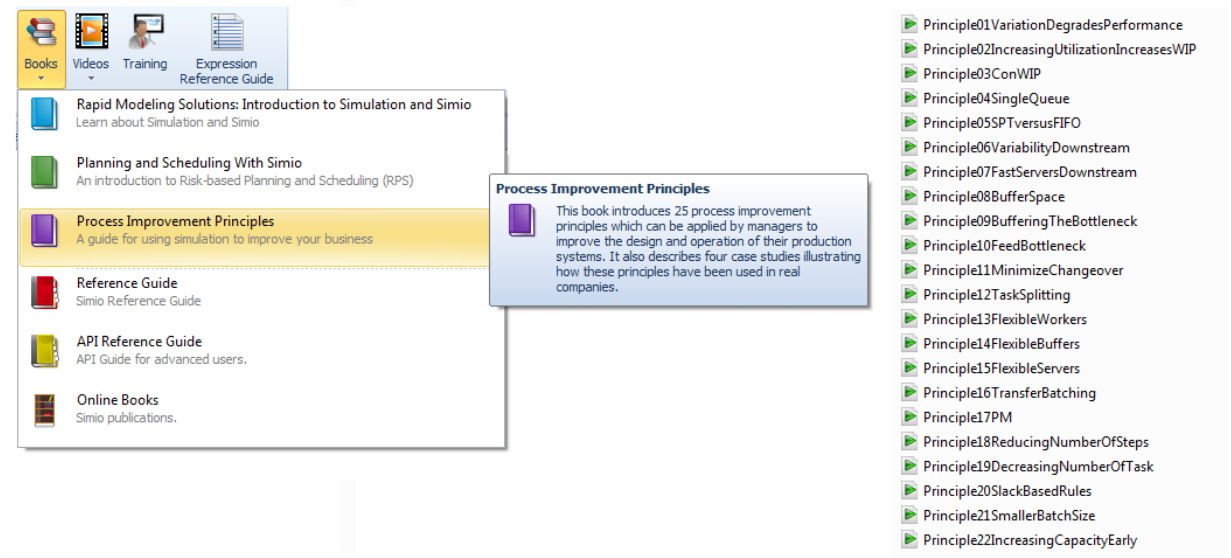
The buttons for these options appear on the Operational Planning tab to the far right of the ribbon as shown.



Please refer to Search Window help topic in the Simio Reference Guide for more information on searching. Note that if a searched item is intended to be hidden within the Scheduler Edition, then that item may appear in the Search window but the Scheduler user will be unable to access the item.

Process Improvement Principles – Examples Folder

We have added a folder of examples that correspond with the book Process Improvement Principles (found within the Support ribbon / Books button). The ProcessImprovementPrinciples folder can be found within the Examples and it contains 22 small examples that correspond to the ideas discussed in the book.



Simio Release 7 – Sprint 120 – April 5, 2015

As we continue our development work on the new Simio Portal Edition, and application-specific enhancements, we have also added some enhancements to the Release and Allocate steps. These changes provide greater flexibility in resource handling, especially as it relates to the most recently added Task Sequences introduced in Sprint 119.

Release Step Enhancement – *Keep Reserved If* and *Reservation Timeout* properties

The **Release** step now provides a new option that allows an owner object to keep the released resource capacity *reserved* for possible later reuse. Reserved resource capacity is capacity that although unallocated has been set aside for a specific future owner thus preventing its use by anyone else.

Whenever resource capacity becomes available, that capacity cannot be allocated to a seize request waiting in the resource's allocation queue unless that there is either sufficient unreserved capacity available or the owner of the seize request has sufficient reservation(s).

For example, if all of a resource's available capacity has been reserved, a seize request without a reservation will have to wait in the allocation queue until either active reservations for the desired resource have been cancelled or sufficient unreserved capacity has been released or scheduled. Note that seize requests with reservations will be allowed to seize a resource even if ranked in that resource's allocation queue behind requests with no reservations.

When an owner object attempts to seize a resource from a group of candidates (e.g., from a list or from a population of some specific entity type), the preference will be to select resources that the owner has reserved. For example, if an entity is attempting to seize a resource from a list that contains candidates 'Resource1', 'Resource2', and 'Resource3', and the entity has an active reservation for 'Resource2', then the entity will have a preference to select 'Resource2' irrespective of the specified selection goal for the seize attempt.

A resource reservation will be automatically cancelled if one of the following events occurs:

- The *Reservation Timeout* property is used and the specified timeout period expires.
- The reservation owner is an entity that is destroyed.

Note that if the *Reservation Timeout* is specified as the value 'Math.EPSILON' then the resource reservation will be automatically cancelled at the end of the current simulation time step if there has been no immediate re-seize of the resource by the same owner object.

Allocate Step Enhancement

The Allocate step now provides an option to cancel all active reservations for a resource (regardless of who has made the reservations) before attempting to allocate the resource's capacity to any waiting seize requests.

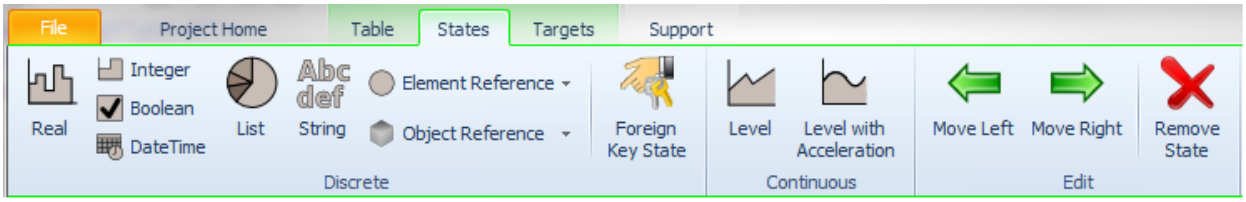
New Resource Related Functions

Resource.Capacity.Reserved – This function returns the current number of capacity units of the resource that have been reserved.

Resource.Capacity.ReservedTo(owner) – This function returns the current number of capacity units of the resource that have been reserved for use by a specified owner object.

Table States (Enterprise)

The states within data and output tables may now be organized by using the new Move Left and Move Right buttons on the States ribbon.



Simio Release 7 – Sprint 119 – March 17, 2015

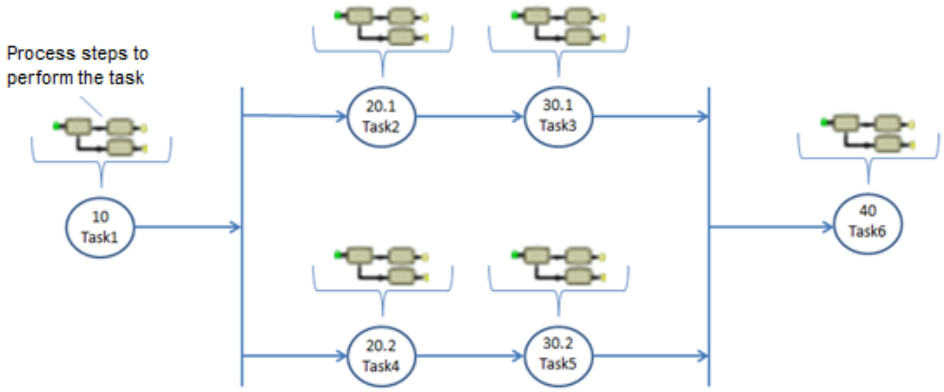
We continue to dedicate much of our development resources into longer-term activities such as planning and scheduling features, the new Simio Portal Edition, and some fundamental features for more powerful modeling capabilities and application development. We are starting to show a 'peek' at those application-specific enhancements through the Task Sequence element, StartTasks step, and integration of those into our Standard Library Server. We have also continued the recent efforts to improve execution speed.

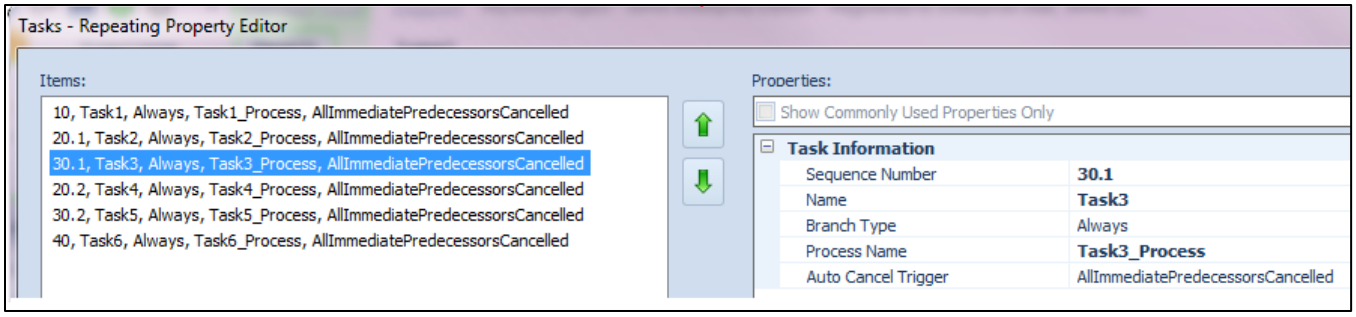
New Element / Step - Task Sequence Element and StartTasks Step

A new element, the Task Sequence element, is used to define and execute structured sequences of processing tasks. This is particularly useful when modeling a set of related tasks which may have parallel and/or serial processing constraints. A StartTasks step references a Task Sequence element and is used to start a task sequence that is associated with an object in the system.
Process flows controlled by a **Task Sequence** element can include serial (one task following another) or parallel (concurrent) processing strategies, as well as conditional or probabilistic branching. When a task is started, you define the action(s) required to perform the task by specifying the name of a process to be executed. A task is considered finished once the original token executing the invoked process has ended.

The following illustrates an example workflow that consists of a collection of tasks which must be processed in a specific order.

Sequence Number	Task Name
10	Task1
20.1	Task2
30.1	Task3
20.2	Task4
30.2	Task5
40	Task6



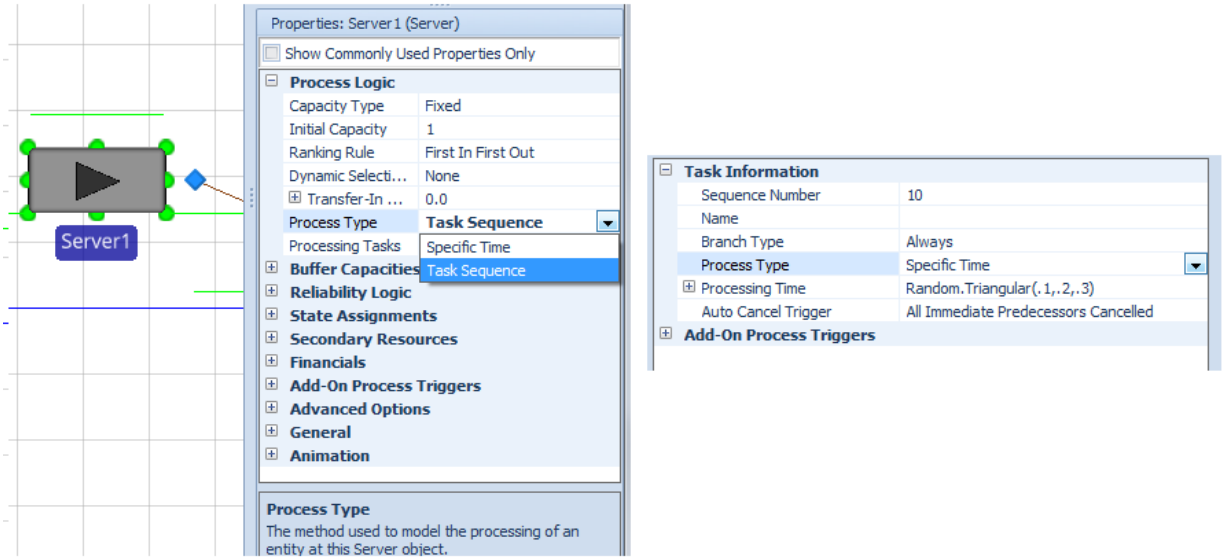


This new feature has also been incorporated into the Standard Library Server object, as discussed below.

Server Enhancement – Addition of Process Type Property and Task Sequence Option

The Standard Library's Server object has been enhanced to include the new *Process Type* property. The options with the *Process Type* property include:

- **Specific Time** – The processing of the entity is defined as a specific processing time. The entity is held at the server until the processing time is completed. ** Server's original behavior **
- **Task Sequence** – The processing of the entity is defined as a collection of tasks which must be processed in a specific order. The entity is held at the Server until the task sequence is completed.



For a *Process Type* specified as 'Task Sequence', the Server object will allow you to model the processing of a set of individual tasks, each using one of the following general process types:

- **Specific Time –** The processing of the task is defined as a specific processing time. The task is considered finished once the time duration has elapsed.
- **Process Name** – The processing of the task is handled by executing a specified process name. The task is considered finished once the token that was created to execute the invoked process has ended its processing.

- **Submodel** – The processing of the task is handled by executing a part of the overall model that is defined elsewhere in the Facility window. The task creates a new entity of a specified entity type and then sends that new entity to a specified node (the starting point for the submodel). The task is considered finished once the entity that was created to execute the submodel has ended its processing (i.e., has been destroyed).

New SimBit

We have added a new SimBit to our library of small examples to illustrate the new features of the Server Task Sequence options.

ServerUsingTaskSequence.spfx - This project includes three separate models that demonstrate the various ways of using the task sequence features to execute a series of tasks from within a Server object.

Execution Speed

We have changed our architecture to support a new type of step-level optimization. We used those new tools to improve the execution speed of seizing resources – in some models this operation may be eight times as fast. Now that this tool set is available, in future releases we will continue to apply it to other frequently used steps.

Data Initialization

Multiple relational data tables will now import much more quickly in certain cases. Parent tables are now automatically imported before child tables, avoiding error handling that would slow the import.

Data Table UI (Enterprise Edition)

New controls have been added to data tables that allow you to move state columns left and right to get them in the exact order that is most convenient.

API Support for Output Tables (Enterprise Edition)

We have added the support for having access to the output table state data through the API. The *StateRows* property of the ITable interface was added to allow reading of table state values. See the API Reference Guide for more details.

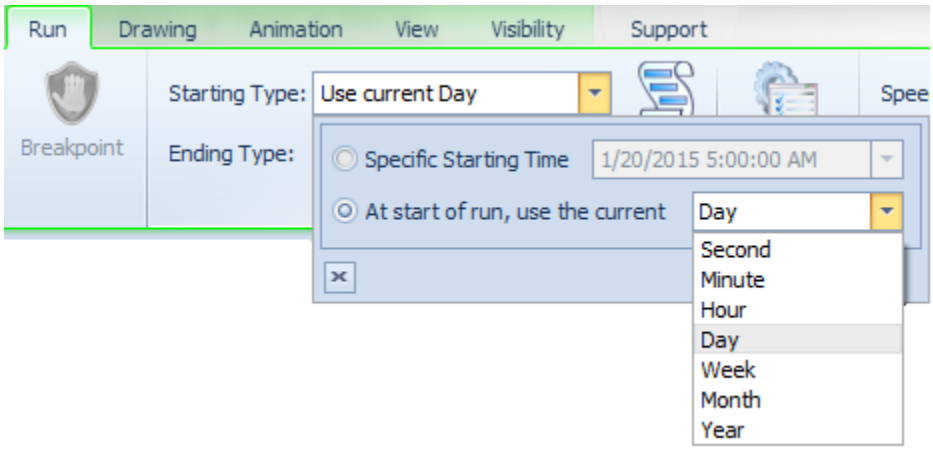
Simio Release 7 – Sprint 118 – February 24, 2015

We have recently dedicated much of our development resources into longer-term activities such as planning and scheduling features, the new Simio Portal Edition, and application-specific enhancements. You will see those appear in coming sprints. The features of immediate benefit include several user requests and improvements to software performance.

Run Setup – Starting Time Enhancement

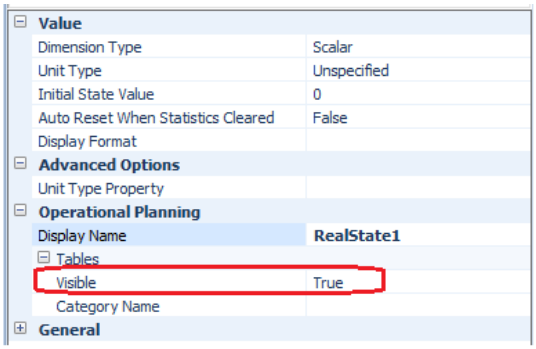
We have enhanced the Starting Time portion of the Run Setup within the Run ribbon. This starting time previously required a particular date and time to start the simulation run. The property name has now changed to *Starting Type* and allows for 'Specific Starting Time', which is the default and only previous behavior. It also has the option to 'At start of run, use the current *' where the user can specify current day, week, month, etc. This is particularly useful in applications like planning and scheduling, where you want to make a series of runs all for the current day or week for example.

Please note that particularly when using such small times as Second, Minute, or Hour, and running the simulation, you may get different results if you are using such features as Arrival Tables and/or resource Schedules, as those are date/time specific.



States within Tables – Enterprise Edition Enhancement

We have added the *Visible* property to the States columns within both Data Tables and Output Tables. Within the Data Tables, standard type columns have had a property named *Visible* that allows an Enterprise edition user to determine whether the column is visible to the Scheduler within the Operational Planning mode (both in the Tables and Gantt right side area). This Visible option is now available as well with any state type columns that are specified either within a Data Table or Output Table.



Performance Improvements

We have made some performance improvements to increase the speed performance of particular areas within some models, as well as corrected some performance issues that were inadvertently introduced in recent sprints.

Simio Release 7 – Sprint 113/114 – November 17, 2014

We are pleased to bring you Version 7 – a compilation of all the features incrementally introduced since version 6. In this release, we have enhanced the Travel and Unbatch steps and the Separator and Pipe objects. We have also opened up Enterprise a bit so you can see what you may be missing. And we have added two new e-books to the software!

Important Notice

In sprint 114, we now automatically install .NET 4.5. For those using an older version of .NET, this will provide a significant speedup to Simio execution. Unfortunately Windows XP does not support .NET 4.5, so Simio 7.114 and later no longer support Windows XP. If you are using Windows XP we recommend installing 7.113, otherwise, we recommend installing 7.114 or later.

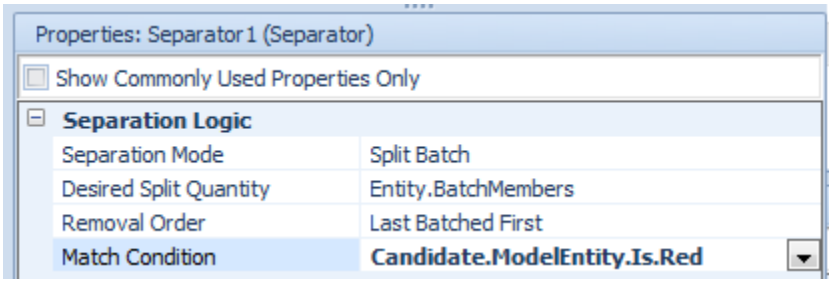
Sketchup 2015 Support

We've enhanced our software so that we now support this latest version of Sketchup 2015.

UnBatch Step and Separate Object Enhancements

We have enhanced the UnBatch step (and related Separate object from the Standard Library) to allow specific entities to be unbatched from a parent entity. Two new properties have been added including the *Removal Order* and the *Match Condition*. If the Desired Split Quantity is less than the number of total batch members, the *Removal Order* will first determine whether the first batch entities or the last batched entities will be removed. Additionally, the Match Condition is an optional expression that can be used to filter the entities. Only the members that match the criteria given by the expression will be split off from the batch.

This is very useful if you have a batch of multiple entity types and would like to unbatch particular entities at various locations. For example, if the batch includes Red, Green and Blue parts, you may use the Match Condition to split off Red parts at one location, Green parts at another location and Blue parts at a third location.



Pipe Changes

We have updated the new Pipe object within the Flow Library to allow splitting at a FlowNode to multiple Pipe objects. In the prior sprint 112, there were some issues with this feature, as noted in the Compatibility Notes which have now been addressed.

Travel Step Enhancements

The Travel step has been further enhanced. In addition to improving the behavior of the new rules, we have also added properties to fine tune that behavior. Most important is that the Steering Rules for the Travel step are implemented in user code. More advanced users can use the example C# code found in the Examples\UserExtensions folder to implement their own, more complex or customized rules. We expect that you will see additional rules supplied by Simio, but you can implement your favorite or customized control behavior yourself, either for competitive advantage or (we hope) to share with the Simio user community.

New SimBit

We've added a new example to our SimBits library:

TravelWithSteeringBehavior.spfx – This example demonstrates the use of the various options on the Travel step for more flexible modeling of entity movement in free space. These features are generally used when you want to show more realistic movement of pedestrians and people. These concepts are often used in agent-based modeling.

Enterprise Edition Enhancement & Documentation

Now everyone can explore Enterprise Edition features. The existing RPSExample has been enhanced, and now can be loaded and fully explored regardless of the activation you have – even in evaluation mode.

You can also find a new *Planning and Scheduling with Simio* e-book that is available under the Support Ribbon Books icon. This 31 page document introduces the planning, scheduling, and risk analysis features and walks you through understanding and running the RSPExample model.

New Book - *Process Improvement Principles*

The objective of *Process Improvement Principles* by Dr. C. Dennis Pegden is to present a set of process improvement principles that can be applied by managers to improve the design and operation of all types of systems – from manufacturing production to health care and distribution. Pegden has boiled down his many years of experience solving "production" problems into a concise set of 25 principles targeted at helping to improve the design and operation of these systems.

This book does not propose or present a process improvement methodology such as Six Sigma or Lean. Instead it presents and illustrates basic principles for process improvement that can be used with any methodology within your process improvement projects.

This book is targeted at busy managers and therefore is intentionally concise. The goal of this book is to convey key process improvement principles in as few words as possible. While this book is sold for $30 on-line, we have an "in" with the author, so we have arranged to include it with Simio.

Simio Release 6 – Sprint 112 – October 20, 2014

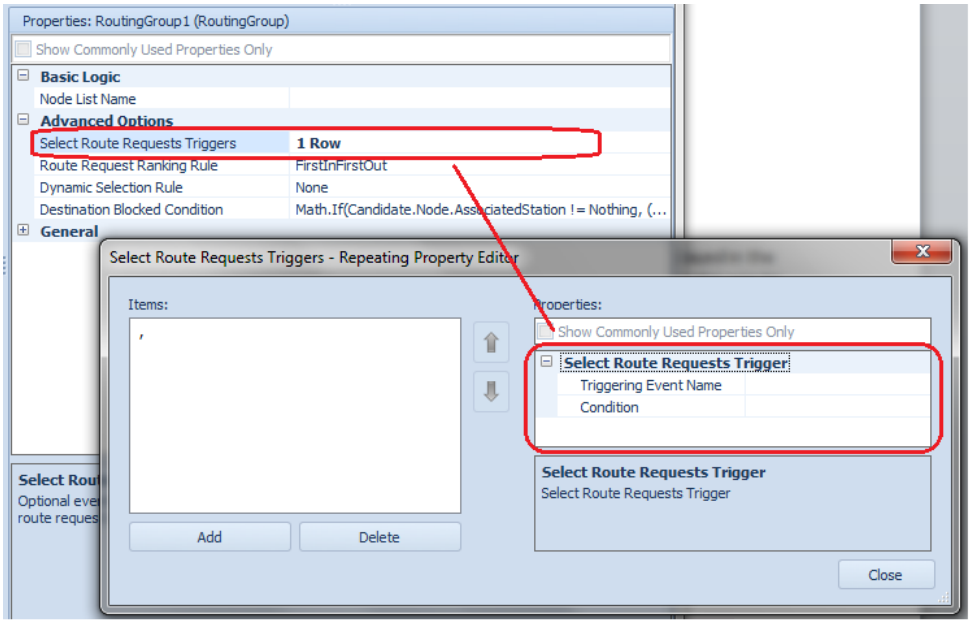
In this sprint, we've added several enhancements based on user requests, including the ability to trigger routing group re-evaluation based on an event(s) in the system, the addition of new rules to the Travel step for more realistic people and entity movement and some enhancements to the Resource Plan Gantt charts.

Routing Group Element Enhancement

We have enhanced the Routing Group element to include a repeating group of Triggering Event Names/Conditions that may be used to cause the re-evaluation of the routing group's route request queue.

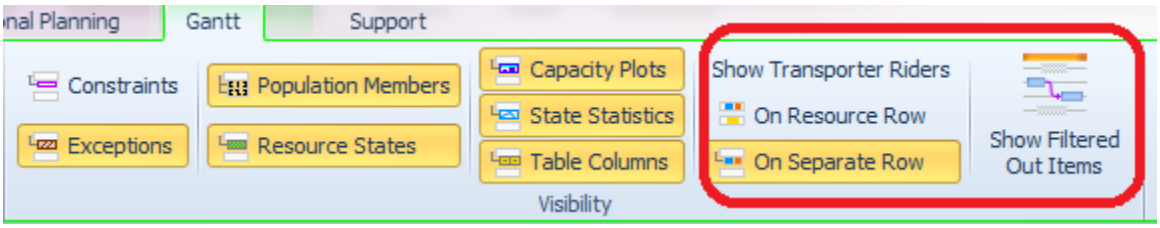
By default, for each candidate destination node in a Routing Group element's node list that is an 'Input' node into an associated object station location, the Routing Group automatically listens for station state changes (such as entities entering or exiting the station or the station's capacity changing) and then does a selection search through its RouteRequestQueue. If you look at the default Destination Blocked Condition that is specified for the Routing Group element, you'll notice that our default definition of blocked is based on a candidate node being an 'Input' node with an associated station and it is considered 'blocked' if the associated station does not have any remaining capacity.

This typically works fine for modeling with our Standard Library objects; however there can be modeling situations where just listening for station state changes at 'Input' nodes is not enough triggering events for the Routing Group to cover all possible selection search scenarios. Per several customer requests, we added this additional selection search to be triggered based on a conditional upstream system event (i.e., number of entities waiting on a path getting larger than 3, etc.).



Resource Plan Enhancements (Simio Enterprise Edition)

We have added new capabilities for the user to decide what is displayed and how it is displayed in the Resource Plan Gantt. In the Visibility section of the Gantt ribbon, there is now a section for the user to decide how entities, that are riders on a transporter, will appear within the Gantt. By default, the riders will appear in the same row as the transporter resource. However, the user can elect to have the riders appear on a separate row, underneath the transporter. Additionally, the user can deselect both options so that riders do not appear at all in the Resource Plan.



The Show Filtered Out Items option has been added to provide users with the option on whether or not to display items that have been filtered out. If the user selects this option to have the items displayed, they will appear in the Gantt in a slightly muted color so that they unfiltered items are easier to see. Otherwise, by default, the filtered out items are not displayed in the Gantt.

Travel Step Enhancements

The Travel step now includes a new property named *Steering Behavior* to provide additional free space travel options for entities. The option for 'Direct to Destination' is the default behavior that the Travel step included previously where you travel to some point (object or coordinates) with some acceleration if desired.

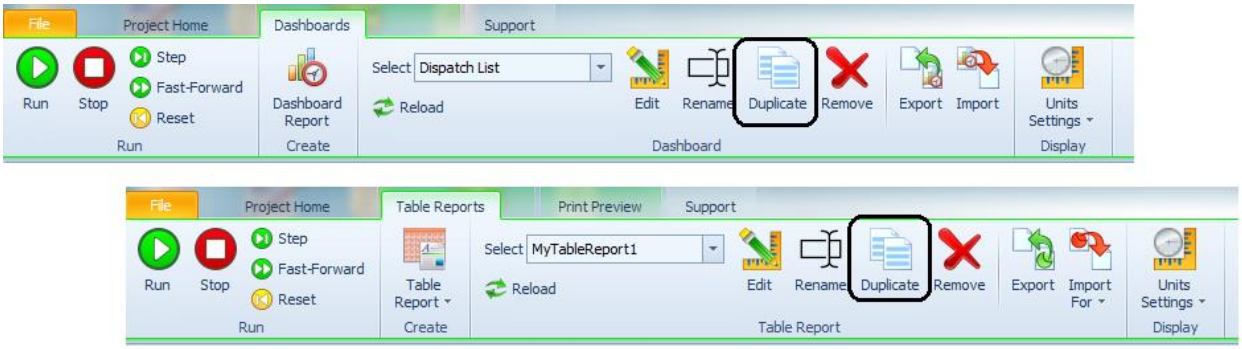
Other new options include 'Follow Network Path' and 'Wander'. 'Follow Network Path' requires a starting and ending node. The entity will then follow the shortest path between them in free space, attempting to stay within the boundaries of the links, defined by each link's Width (or 1 meter if no Width is defined). The Path Width can also be varied so that the entity can directly specify a radius that will override the link's width. 'Wander' simply has the entity wander around endlessly.

Simio Release 6 – Sprint 111 – September 26, 2014

In this sprint, we've added new report capability and several new SimBits to display various software features that we've used. We've also added several new link functions, one in particular which a number of customers have requested – the Link.Contents which is the queue of entities that are located on a link. We've also added some other user requested features, as noted below.

Dashboard / Table Reports Enhancement

Within both the Dashboard Reports, as well as the Table Reports, we have added the ability to Duplicate a particular report as shown on the ribbons below. This will make it easier to create a new report similar to an existing report.



New Link Functions / States

We have added three new states and a new function to the Link type objects:

Link.Contents – This state accesses the queue of entities (ranked in order of entry) that are physically located on the link. This is particularly useful when you are doing more advanced link logic, for example examining a link for available space or an accumulation.
Link.CrossSectionalArea Function – This function returns the cross-sectional area of the link in square meters.
Link.AvailableVolumeFlowInRate State – This state gets the current potential volume inflow rate for the link in the ideal case where there are no link-related flow constraints. Note that this is a private variable primarily intended for flow link object builders (and thus will only be visible inside a link object's process view).
Link.AvailableWeightFlowInRate State – This state gets the current potential weight inflow rate for the link in the ideal case where there are no link-related flow constraints. Note that this is a private variable primarily intended for flow link object builders (and thus will only be visible inside a link object's process view).

Facility Window - Grid Enhancement

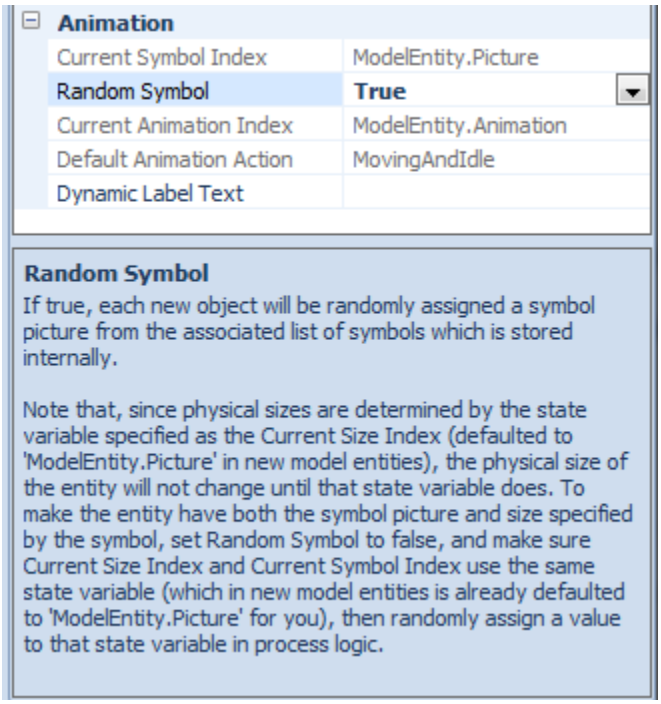
On the grid now if you select different display units, instead of always drawing the gridlines at multiples of 10 meters apart, and displaying the converted value there (like 3.208 feet), we now draw the grid lines at multiples of 10 of the display units specified, so now you'll see 0.2 feet, 2 feet, 20 feet, etc.

Sketchup 2014 Support

We've enhanced our software so that we now support this latest version of Sketchup 2014.

Animation - Random Symbols Update

When the Random Symbols property is set to 'True' for entities, now the Current Symbol Index property will be grayed out. If this value is true, all entities will have the same physical size, based on the entity size of the first entity in the active symbol list (value of '0'). We have also enhanced the help to more clearly describe the behavior and recommended approach for random symbols of different sizes.



New SimBits

We've added four new examples to our SimBits library:

TransporterListForVehiclesOnDifferentSchedules.spfx – This example demonstrates the use of a TranporterList when using various schedules (first shift / second shift) for selecting a vehicle. This could be applicable to workers / resources as well. Two vehicles have different schedules and the vehicles are put into a List. Entities select the vehicle for transport based on the list, which will incorporate their scheduled availability.

SetPropertyBasedOnValueOfListProperty.spfx – This model uses a string list property and based on the value of the property in the list, the model will set a different model property numeric value using a Math.If statement. This is useful for customers that want to select control type information based on a string value, while that value is used to then set up other parameters in the system.

SearchStationElement.spfx – This model uses a Station element queue to hold entities in one area of the simulation. Another area of logic then searches the station element queue contents for a matching type entity based on a specified match condition. If a matching entity is found in the station queue, it will be removed and batched with the searching entity.

ScheduleUsesAProperty.spfx – In this new SimBit, a work schedule for a server is used, where the capacity within the schedule is dependent on a user input property. Therefore, the server's capacity goes between '0' (off shift) and a property value for capacity (on shift).

Simio Enterprise - Gantt Charts

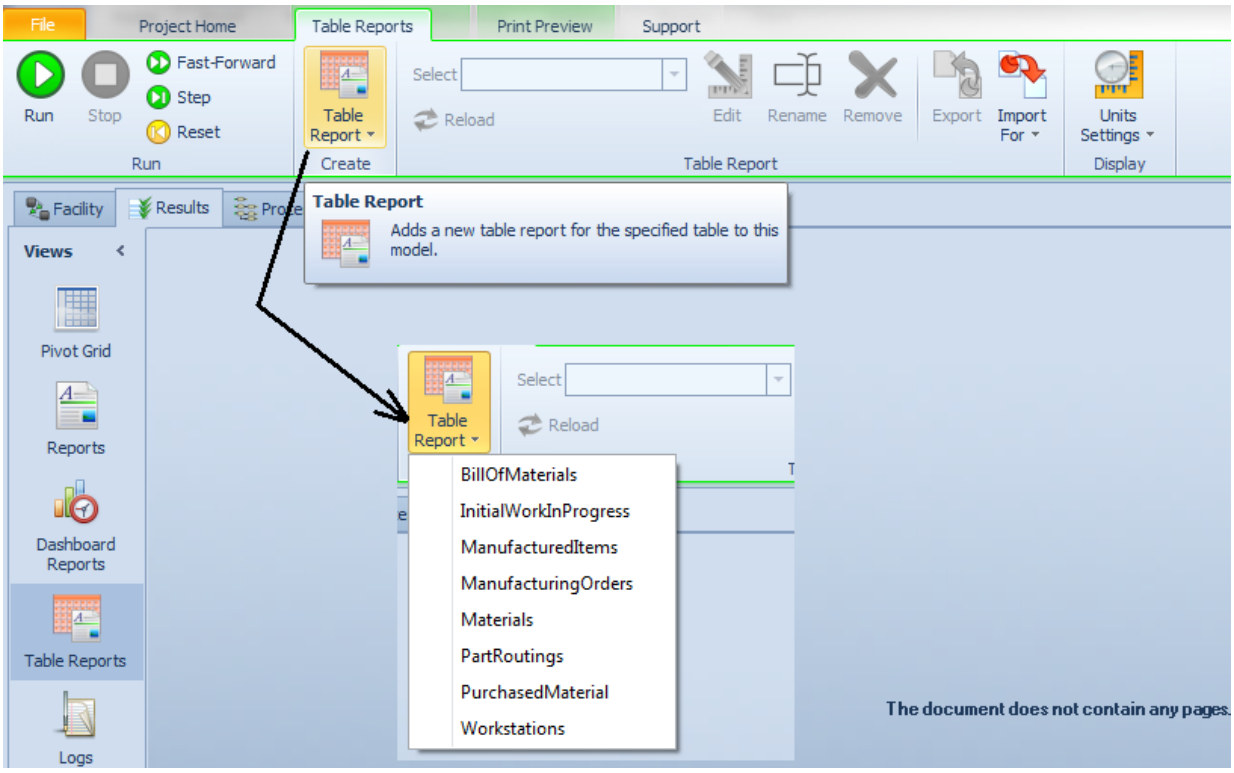
Entity and Resource Gantt Chart settings, such as View, Zoom levels, Targets and Visibility settings are now saved with the project.

Simio Release 6 – Sprint 110 – September 9, 2014

In this sprint, we have enhanced our reporting ability by adding several features, in part to support concurrent model development. And we have simplified the expression editor to make it easier to find the items you want. We've also continued to provide additional SimBits based on new features and modeling ideas.

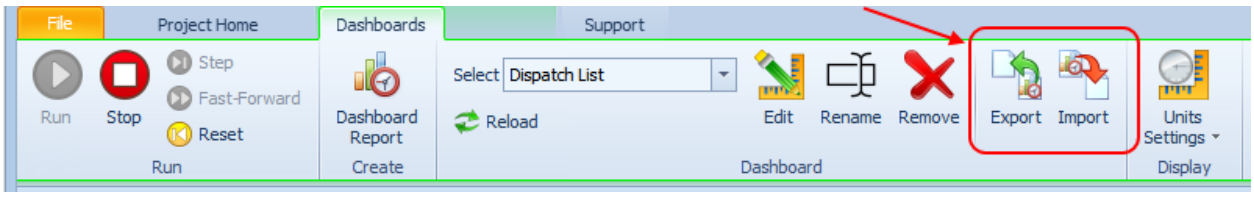
Table Reports

Within the Reports tab of a model, you may now create your own Table Reports based on the tables in the Data window (Enterprise edition) or view previously generated reports (all users). As shown in the example below, the tables within a model are available for selection within the Create - Table Report button. Selecting a table will then open the Report Designer window for the user to generate custom reports.



Dashboard Reports

We have added the option to Import and/or Export Dashboard Reports. The format is XML. This feature is intended for use when moving dashboards to or from other projects (for parallel development).

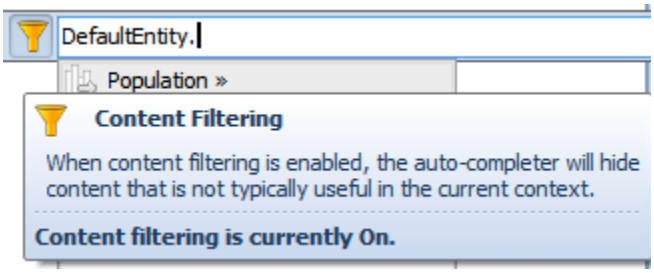


Expression Editor Content Filter

A new filter button on the left side of the expression editor allows you to limit the list to a context-determined subset of items.

When in the main model, the filter hides all of the statistic items (making a slightly shorter list). The big payoff is when you create the expressions for experiment responses the filtered list shows only statistic items, which are often the only items used. This makes experiment expressions MUCH easier to create. In either case, if you want to see the full list that you used to see, just disable the filter.

In a related change, the expression editor will now behave more intuitively when you are typing more complex expressions, especially those involving items that are not shown on the list.



New SimBits

We've added a two new SimBits to our SimBits library of examples:

CombinerNode.spfx – This model uses a custom node object that batches multiple entities of given 'priority' values together to move to the next location in the model. The Batch step and BatchLogic element are used to provide combiner-like functionality to a node object.

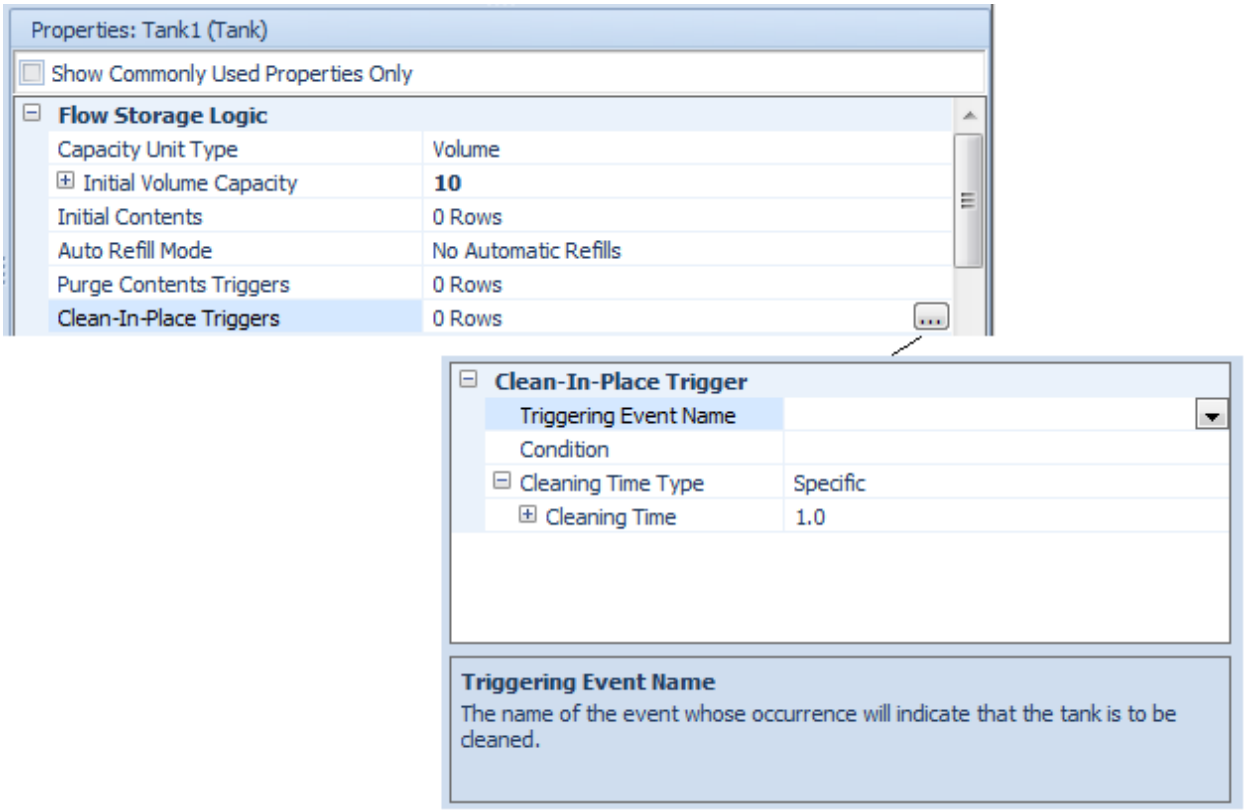
TankCleanInPlaceTrigger.spfx – This model demonstrates the use of clean in-place triggers that were added to tanks (and other converters) in sprint 109. This example includes three types of entity flows entering a tank. When a new product flow type is used, the clean in-place triggers start the cleaning of the tank (with changeover matrix information) before the new product flow type begins.

Simio Release 6 – Sprint 109 – August 15, 2014

In this sprint, we have enhanced several of the Flow Library tank based objects to include Clean In-Place Triggers and fixed a number of issues found by someone using fairly advanced Flow configurations. So the Flow library is now both more feature-rich and more robust.

Flow Library – Tank, FlowToItemConverter, ItemToFlowConverter to Include Clean In-Place Triggers

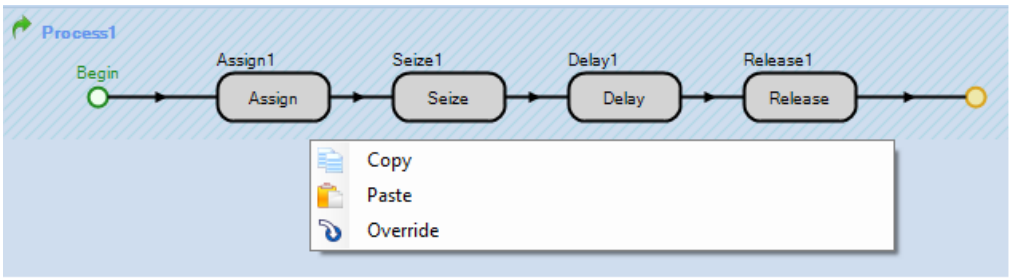
We have enhanced the three Flow Library objects that include built in Tank objects (Tank, FlowToItemConverter and ItemToFlowConverter) to include optional event driven triggers that will immediately remove and dispose of any contents held in the tank and then suspend new inflow until a specified cleaning time has elapsed.



In conjunction with these tank based changes, the graphic symbols for new objects placed with an imbedded tank will include the states of the object. Tank, FlowToItemConverter, and ItemToFlowConverter objects will be gray (default) for the Empty (0) state, green for the PartiallyFull (1) and Full(2) states and yellow for the Cleaning (3) state.

Processes Window – Right –Click Process to Override/Restore

We have added the capability through a right-click on an inherited process to either Override or Restore the process. This functionality has been available only through the Process ribbon in past sprints. This was a user-requested feature.


Process1
Begin
Assign1
Assign
Seize1
Seize
Delay1
Delay
Release1
Release
Copy
Paste
Override

Simio Release 6 – Sprint 107- 108 – July 25, 2014

In these past several weeks since we officially posted sprint 106, we've added a number of user requested features, including fonts for floor labels, enhanced spreadsheet editing and expression editor, and an option to reset state values when clearing statistics. These and more are described below.

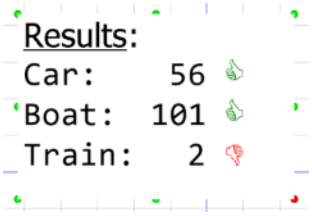
Floor Label Enhancements - Different Fonts and Proportional/Fixed Buttons for Default Font

You can now specify the font (or fonts) you want in a floor label with thetag:

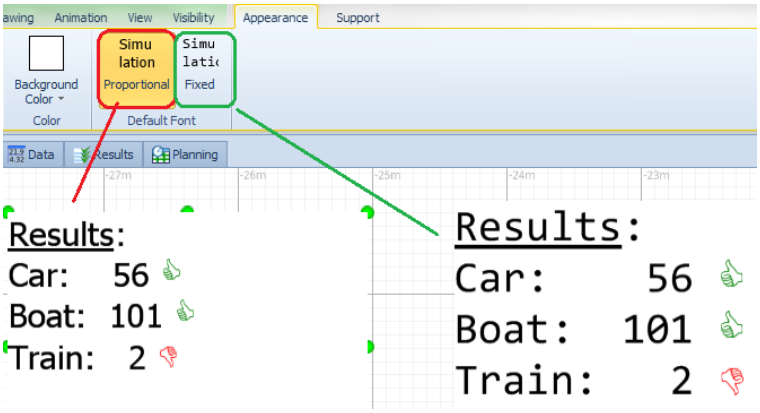
So for this markup:

<u>Results</u>:
Car: 56 <green>C</green>
Boat: 101 <green>C</green>
Train: 2 <red>D</red>

You get this result:



Alternatively, if you are using the Default Font (not specifying a particular font as noted above), you can now choose the Default Font 'spacing' for the floor label.

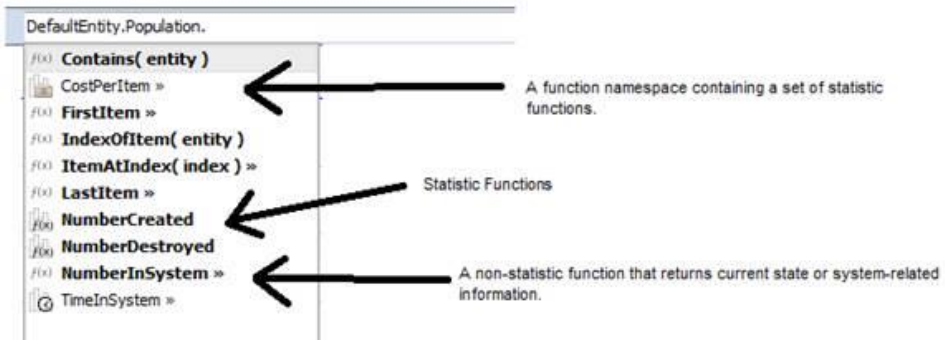


The Proportional and Fixed buttons don't affect the overall text layout, only the text not surrounded by <font> tags. Our default Proportional font is "Tahoma" while our default Fixed font is "Consolas".

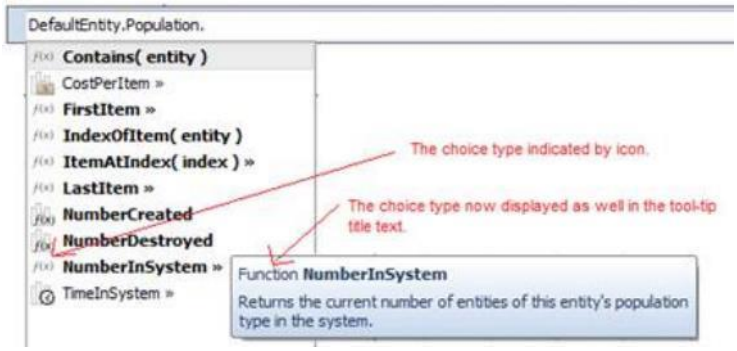
Expression Editor Auto-Completion Choice Related Enhancements

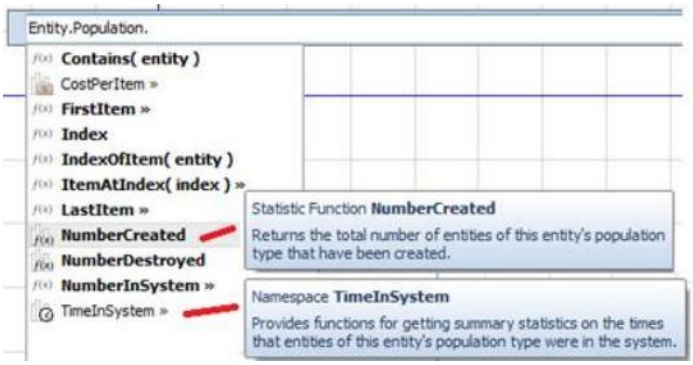
We have made some enhancements to the expression editor auto-completer to distinguish statistic type functions from other functions, as well as to provide additional tool tip information.

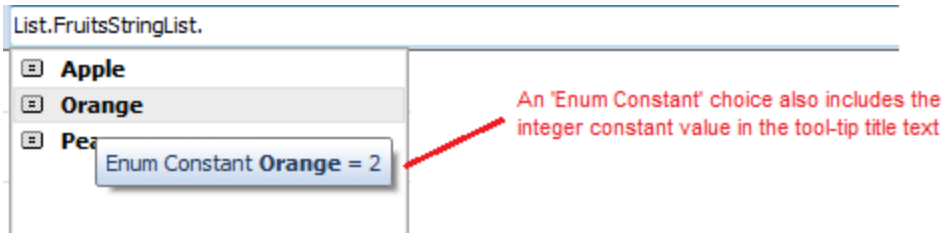
In the expression editor auto-completion choice list, a statistic function choice is now displayed with a different icon than a non-statistic function choice.



In addition to the icon displayed for an expression editor auto-completion choice type in the selection pull-down list, the mouse tool-tip title text for a choice now also includes the type information in textual form.

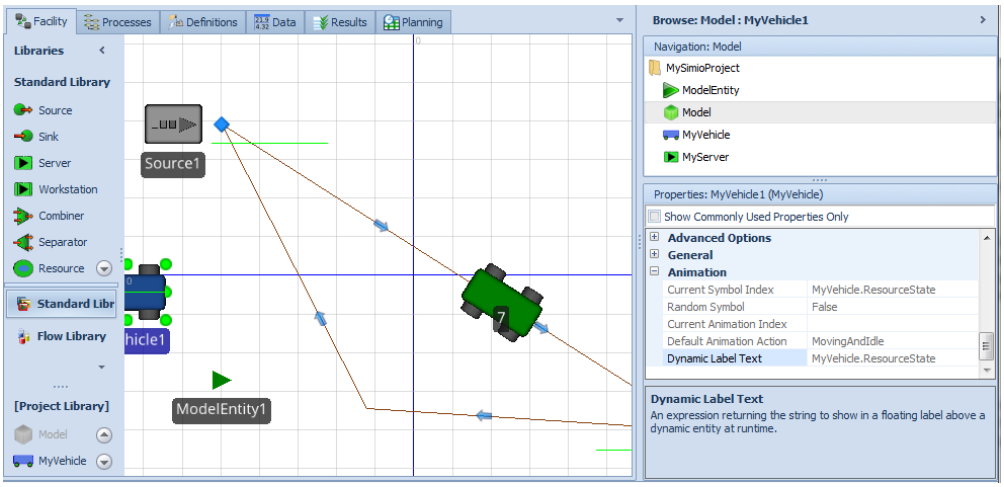






Default Dynamic Label Text

When you go to the External view of an Entity or Transporter and click in the window, you will now see a single property called *Default Dynamic Label Text*, which is the default value for the *Dynamic Label Text* property of instances of the object. Dynamic Labels can be turned on and off using the Visibility ribbon buttons.



Size / Location Object Properties in Spreadsheet Editing

Several months ago, we added the Properties Spreadsheet for easy editing of Facility Window objects within a spreadsheet interface. We've enhanced this feature to include the Size (Length, Width, Height) and Location (X, Y, Z) under the General/Physical Characteristics section of properties. Node object will not include the Size properties and Links will not have the Length property editable. This feature is especially useful when a user is trying to line up multiple objects at the same X location line for example, or resize multiple objects to be the same physical size, as copy / paste to multiple cells can easily be done in the spreadsheet interface.

API Enhancement – Set Starting Time for Time Indexed Tables

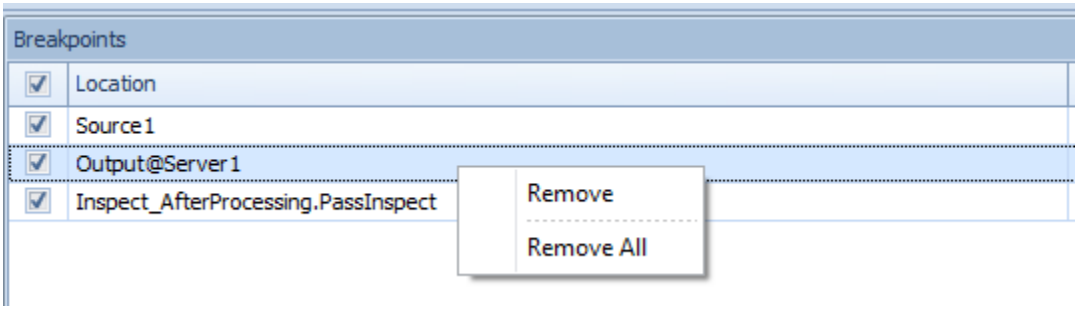
We have added the option for setting the starting time for time indexed tables from within the API. This feature is within the ITable Interface and is referenced as the TimeIndexedStartTime property.

Roaming Licensing with Uninstall Change (*Network License Servers)

For those with network licenses, uninstall of Simio will now fail if you have a roamed license. You must first return the license before uninstall process can begin.

Context Menu in Breakpoints Window for Deleting Breakpoints

We've added a right click context menu within the Breakpoints window to either remove the currently highlighted breakpoint or to remove all breakpoints in a model.

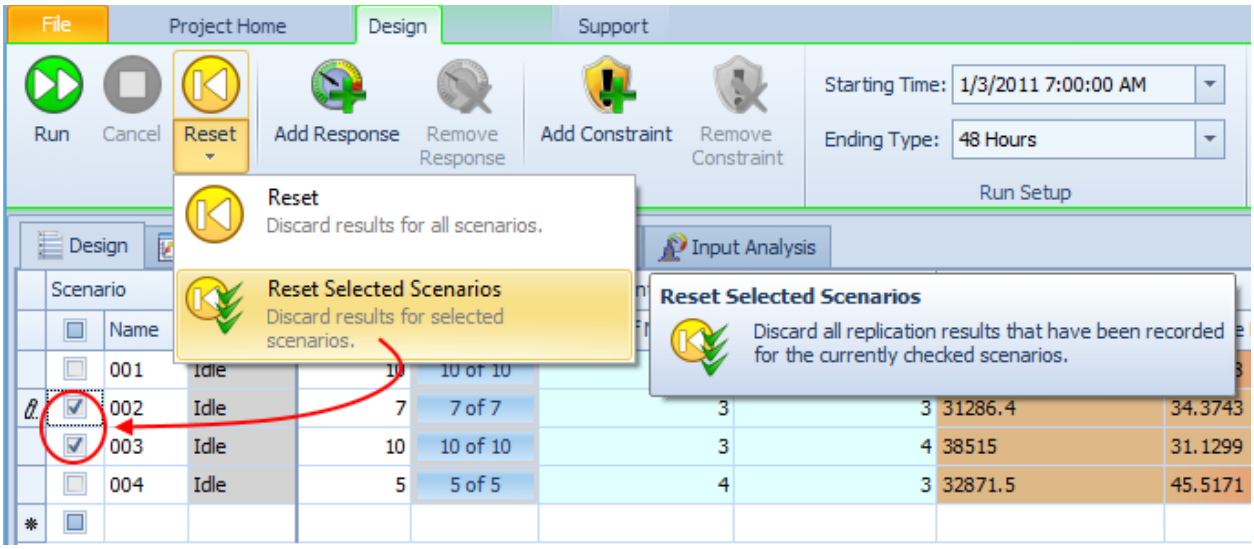


New 'Auto Reset When Statistics Cleared' Option on States

For any state variable definition of any data type and regardless of whether a scalar or an array, in the Value property category we now provide a new *Auto Reset When Statistics Cleared* property. If this option is 'True', then the variable will be automatically reset to its initial state value(s) whenever the parent object or element's statistics are cleared (e.g., after a specified warm-up period in the experiment or due to use of a ClearStatistics step). This feature is especially useful for those users who are using state variables to collect count-based or other statistics in the model.

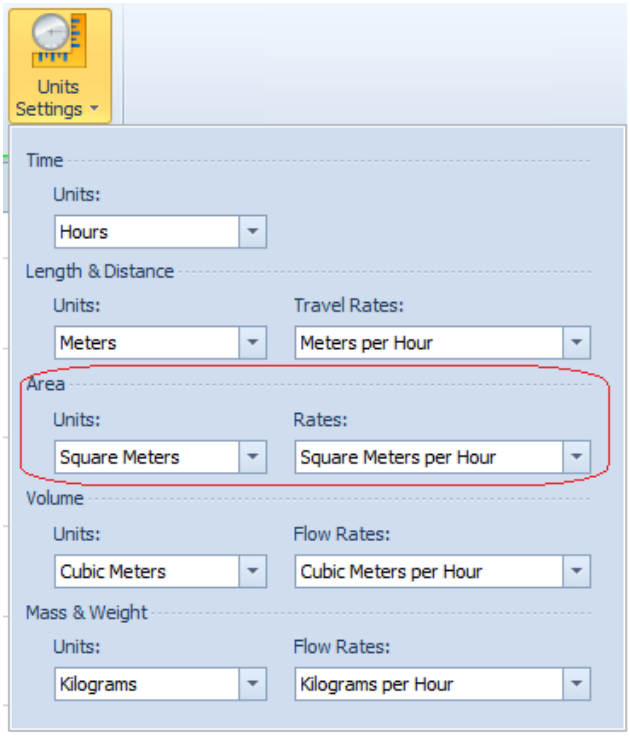
Reset Selected Scenarios Option on Experiments

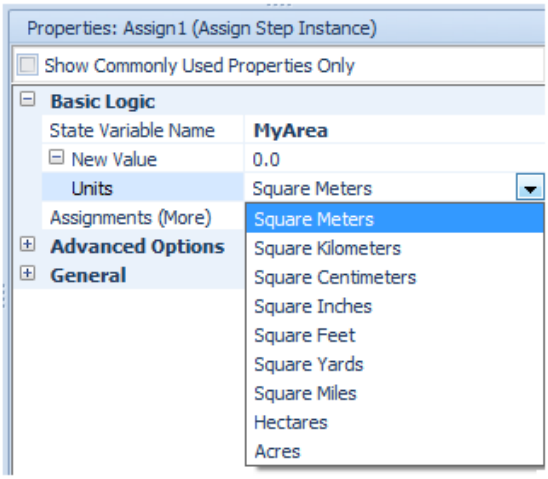
In addition to our normal Reset button, there's now a button for resetting only selected scenarios (i.e. the ones with their check boxes checked).



New 'Area' and 'AreaRate' Unit Types and 'Long Ton' Mass/Weight Type

We have added a new unit type for Area including the individual and rate units. This allows state variables, properties, statistics, trace, etc. to be shown in units of area, such as square meters, square kilometers, square centimeters, square inches and so on as shown below (along with corresponding rate units).

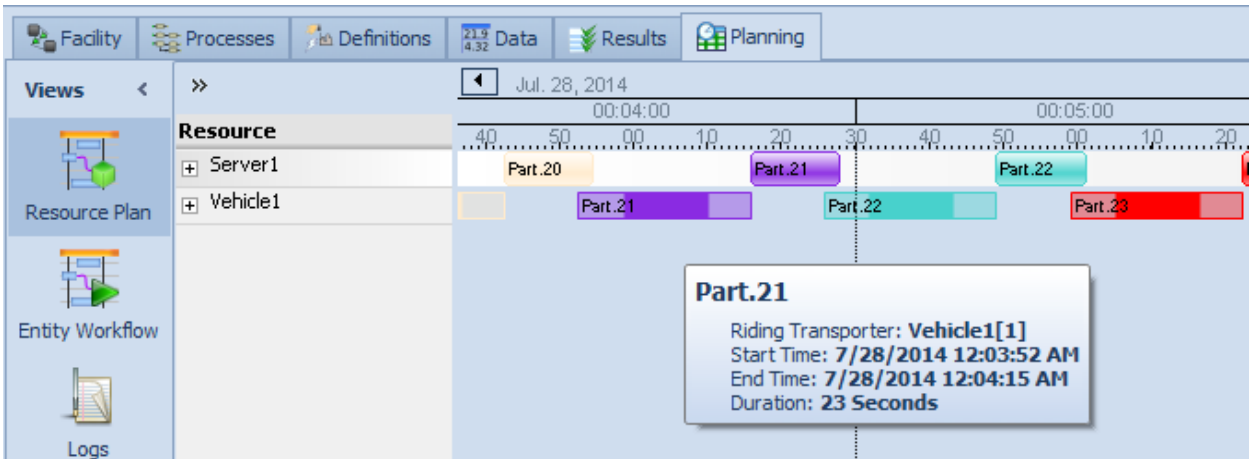




Additionally, within the Weight units, we have added Long Tons (previously had Ton and Metric Ton). The British ton is the Long Ton, which is 2240 pounds, and the U.S. Ton is the Short Ton which is 2000 pounds (this is Simio's Ton).The Metric ton is equal to 1000 kilograms, or approximately 2204 pounds.

Simio Enterprise Edition – Resource Plan Gantt Enhancement for Vehicles / Workers

We have changed and enhanced the way we are displaying the usage time for Vehicles and Workers within the Resource Plan Gantt. In prior versions, we did not include Load Time in the Gantt. Now we display the Load, Transport and Unload, distinguishable by color. The lighter color section on the left indicates the load time, the darker middle section represents the transport movement time and the lighter color section on the right indicates the unload time with the Vehicle or Worker. When you hover over the specific entity, the entire Busy/Transporting duration is displayed (includes load/transport/unload times).

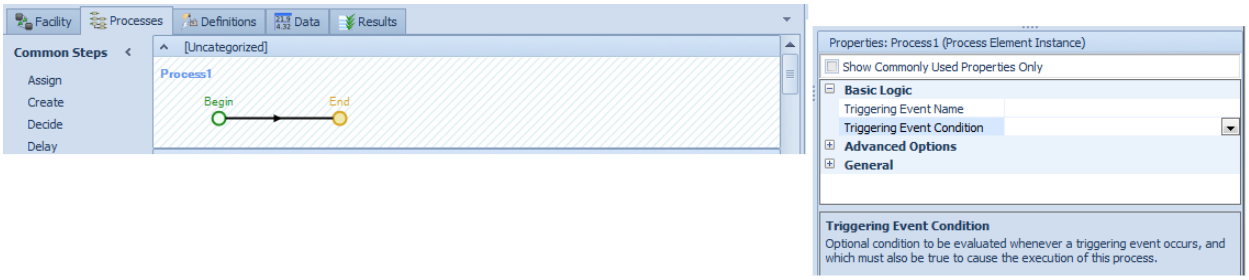

Facility
Processes
Definitions
Data
Results
Planning
Views
Resource Plan
Entity Workflow
Logs
Resource
Server1
Vehicle1
Jul. 28, 2014
00:04:00
00:05:00
Part.20
Part.21
Part.22
Part.21
Part.22
Part.23
Part.21
Riding Transporter: Vehicle1[1]
Start Time: 7/28/2014 12:03:52 AM
End Time: 7/28/2014 12:04:15 AM
Duration: 23 Seconds

Simio Release 6 – Sprint 106 – June 13, 2014

We are now stabilizing for our academic-focused mid-year release. This means that for this and the next sprint we are not adding much in terms of new features, but instead putting extra time into testing and polishing our software and documentation. Here are a few changes you will see:

Processes Window Enhancement – Triggering Event Condition

A new property has been added to a Process within the Processes window. The *Triggering Event Condition* will allow users to specify a condition that must be evaluated to true in order for the process to be triggered by the event.



Simio Network License Manager Features

The administrator can now require Users to log in to the Simio Network Manager administrative web page. Each user has a set of access rights, which define what menu items appear when they log in.

The new INTERNET_GROUP is added to allow creation of groups to segment licenses based on IP addresses.

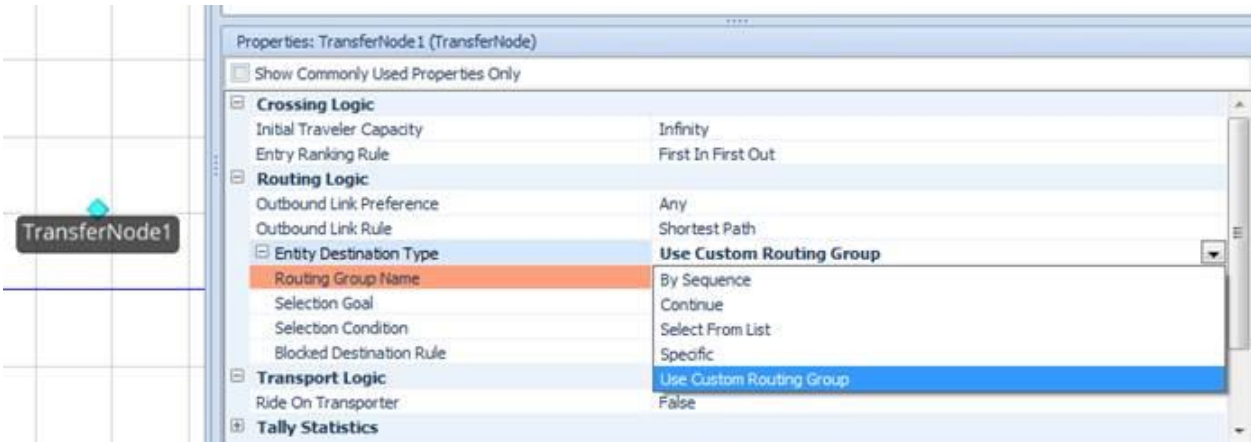
The RLM reportlog now logs all licenses currently in use at the start of the log, and all licenses in use at the end of the log as well.

Changeover Matrix Enhancement

Users can now use the syntax **ChangeoverMatrixPropertyName.Matrix.Name** to get the string name of the changeover matrix that has been specified using the property. This is an additional enhancement to some of the changeover matrix changes made in Simio sprint 104.

Transfer Node – Entity Destination Type Enhancement

Within the TransferNode object (and all objects that include the TransferNode, such as Source, Server, Workstation, etc.), we have added a new option to the *Entity Destination Type* for routing from the node. This property now includes the option to 'Use Custom Routing Group' as shown below:



If you select that choice, the name of the Routing Group element to use is specified. Note that this is an advanced alternative option to route each entity to a destination selected from a list of candidate nodes (versus using the 'Select From List' option).

Use cases for this new option include:

- If you are modelling blocked routing behaviour and have entities waiting at multiple TransferNode objects that are trying to route to a destination selected from a common node list, and you want a common set of decision rules to apply to all of those entities (e.g., entities at one location perhaps prioritized over entities at another location when a destination becomes available, etc.), then this option is available to use.
- If you are modeling blocked routing and want more advanced control over how a waiting entity is selected when a destination becomes available (e.g., using a static ranking rule for the RouteRequestQueue that is not FIFO or using a dynamical selection rule for more advanced 'pulling' of entities), then this option is available to use.
- If you want to customize the conditional expression which indicates whether a destination node in the list of candidate nodes is considered 'blocked, rather than using Simio's default condition, then this option is available to use.

New SimBit

We've added a new SimBit to our SimBits library to support the above mentioned TransferNode enhancement:

CustomRoutingGroup.spfx – This model illustrates the use of the new *Entity Destination Type* option of 'Use Custom Routing Group'. Entities from two sources feed into three servers using the same logical routing group logic so that when a server is available, it 'pulls' from the same queue of waiting entities, even though the entities are physically located at different servers.

Simio Release 6 – Sprint 105 – May 28, 2014

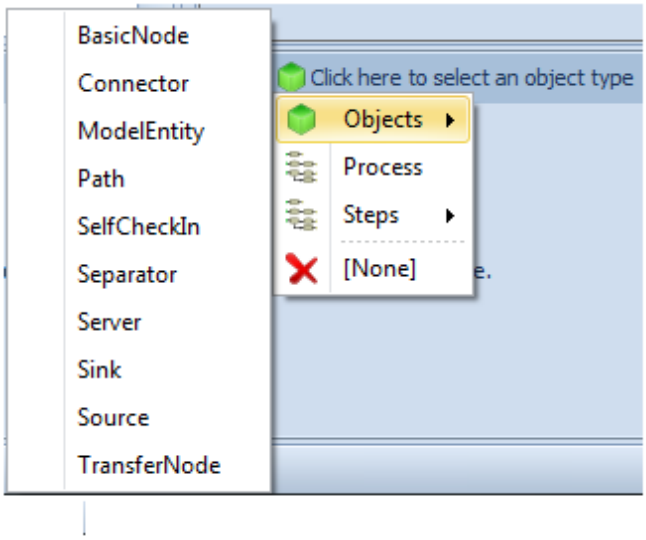
In Simio Sprint 105, we have enhanced the Processes window editing to include the availability of the Properties Spreadsheet window and multi-select for property editing. We've included new functions for entity type and vertices on links, as well as additional API support and many other user requested features.

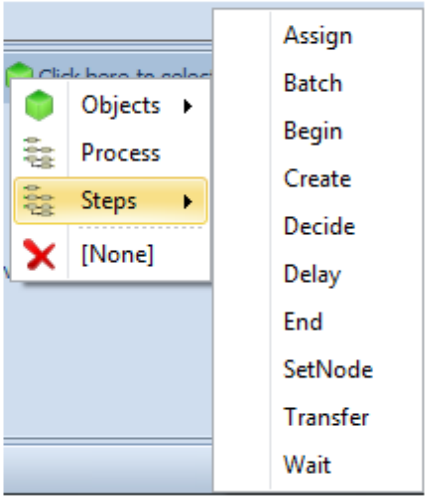
We've also set a new bar for Input Analysis by implementing work by Dr. Barry Nelson and Eunhye Song. We have enhanced the recently introduced Input Parameters with an Input Analysis section of Experiments. Here you can easily and automatically determine how sensitive each of your responses is to your input parameters. You can also optionally extend your analysis to determine how your response error is affected by the data associated with your input parameters. You can even determine the relative value of collecting extra data for each of your input parameters.

Processes Window Enhancements - Properties Spreadsheet and Multi-Select

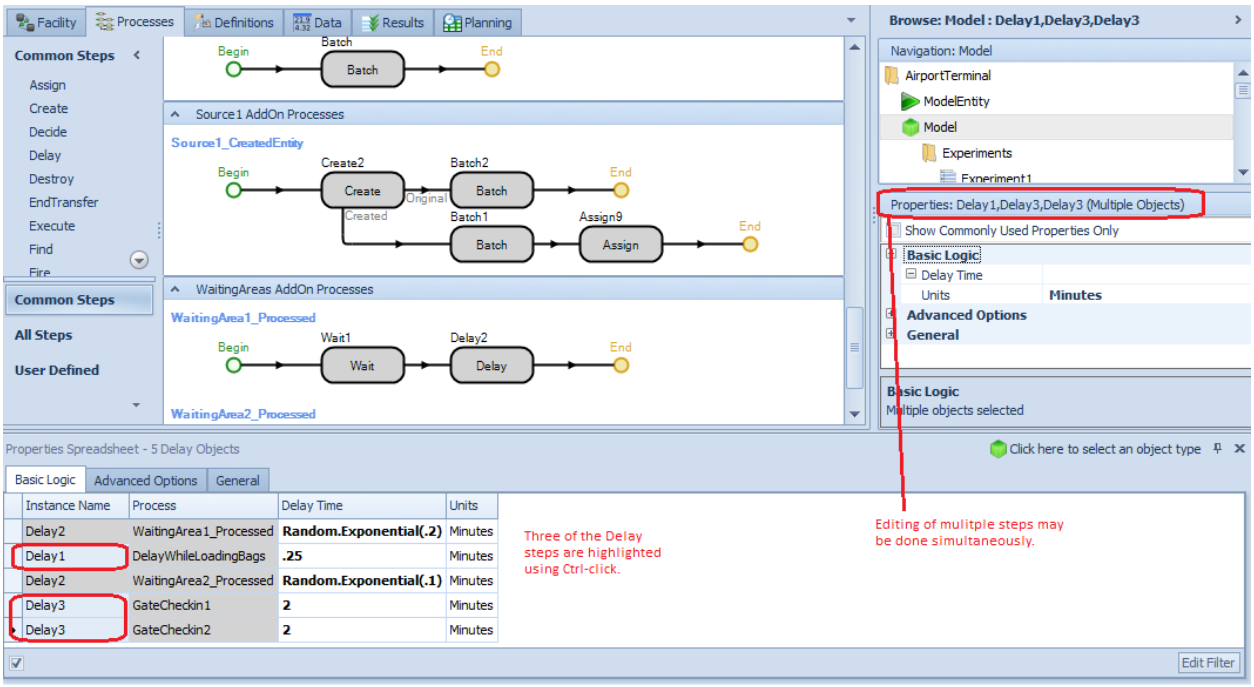
Several months ago in sprint 101, we added a new Properties Spreadsheet window (Properties button on the Project Home ribbon) that allowed users to view and edit the Facility window objects in a spreadsheet format.

We've now enhanced this feature to include the viewing/editing of Processes as well as Steps within Processes window. The 'Click Here to Select an Object Type' now includes Objects, Process and Steps and has additional menus to select the particular object type or step type desired, based on what has been placed in the model.





Editing of multiple steps may now be done through the Properties spreadsheet as shown below. Copy/paste can be used within the grid. Additionally, we've added the multi-select option for selecting multiple steps. This feature allows similar properties to be edited through the Properties window on the right, as shown below.



New Entity Function and Link Functions

Entity.EntityType – Returns a reference to the entity type of this entity.

Possible uses of this function include:

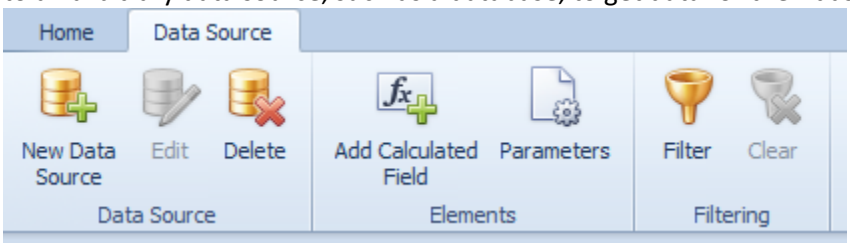
- Referencing it in the Entity Type property of a **Create** step. For example, if you wanted to create a new entity of the same entity type as the first entity located in a station location, you could do **Create** new object with Entity Type = TheStationName.Contents.FirstItem.Entity.EntityType.
- Referencing it as the entity type to search if using the Search step to search an entity population. For example, you might say to Search an entity population with Entity Type = Entity.EntityType to search the entity type of the executing token's associated entity.
- Using it to save an entity type into a state variable, such as the Flow Library Tank perhaps saving the entity type of the last inflow entity into a LastInflowEntityType variable. And then can compare the value of that state variable with the entity types of the next inflow entities to detect when the inflow entity type has changed.
- Referencing the string name of an entity's entity type using the syntax Entity.EntityType.Name.

Link.Vertices.NumberItems – Returns the number of vertices in the link. Note that these are the "interior vertices", meaning the end points of the link are exclueded, since those are already available via the nodes at each end. A straight link between two nodes would return 0 for *NumberItems*.

Link.Vertices.ItemAtIndex(index).X – Returns the current location of the specified vertex along the X axis of the Simio grid.
Link.Vertices.ItemAtIndex(index).Y – Returns the current location of the specified vertex along the Y (vertical) axis of the Simio grid.
Link.Vertices.ItemAtIndex(index).Z – Returns the current location of the specified vertex along the Z axis of the Simio grid.

Dashboard Reports – Data Source Ribbon

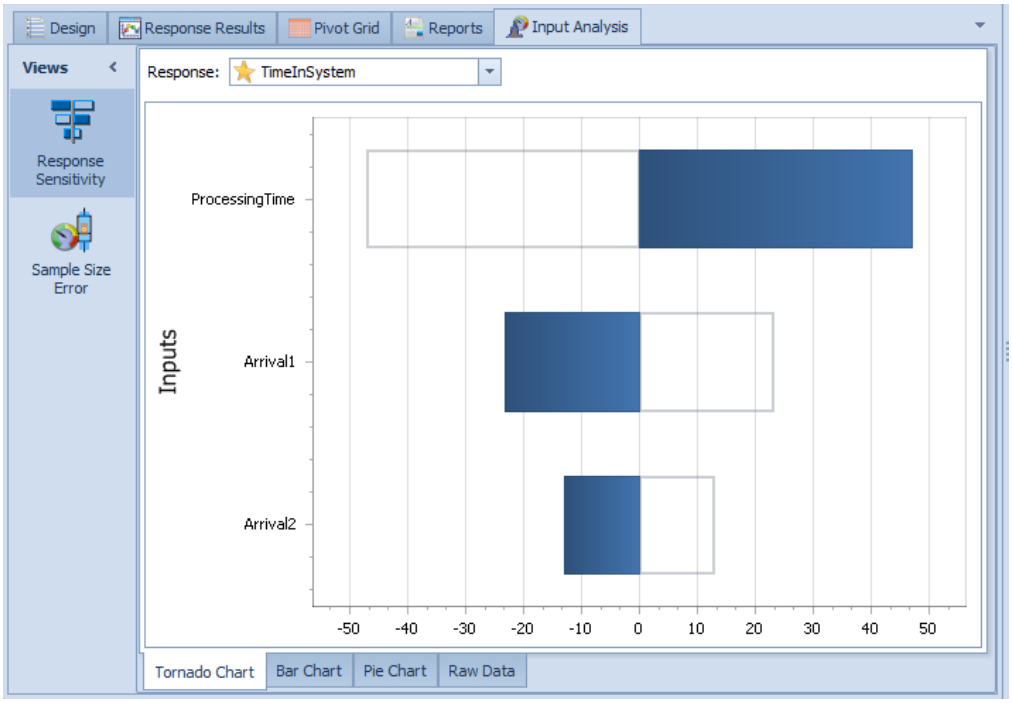
We have added the Data Source ribbon tab in the dashboard reports designer, so now users can connect to an arbitrary data source, such as a database, to get data for their dashboards.



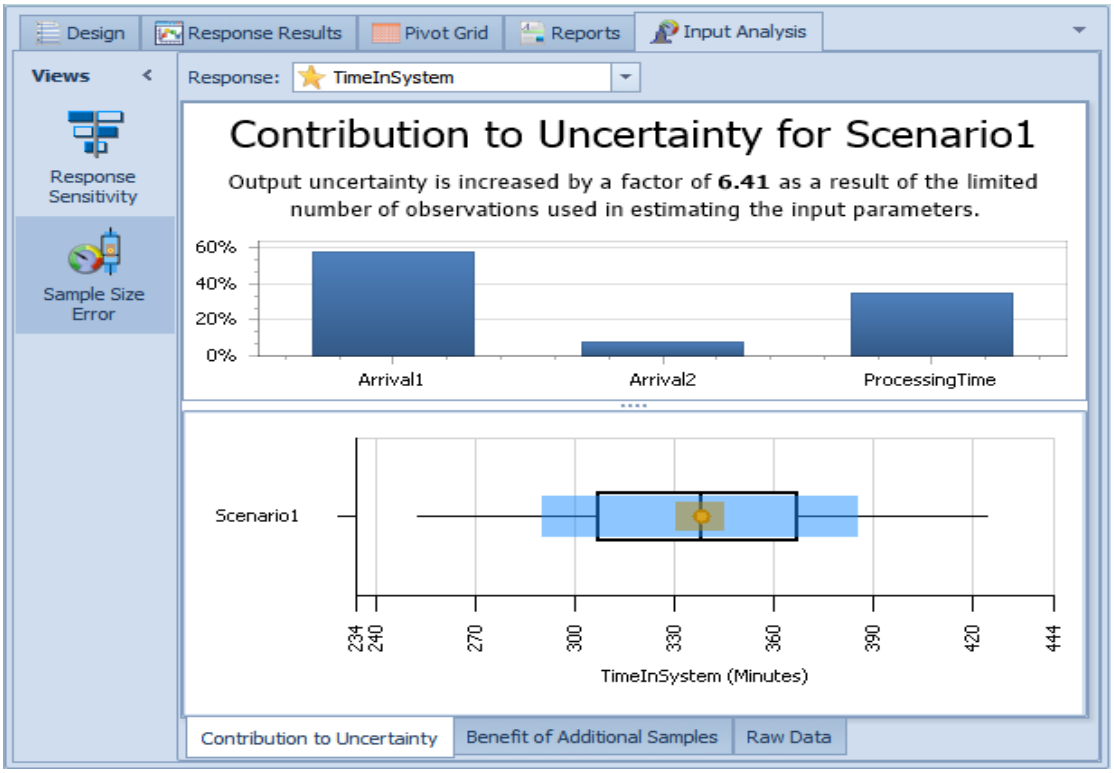
Experiments – Response Sensitivity and Sample Size Error Analysis

We've changed the user interface and terminology for input analysis which was introduced in Simio Sprint 102. Simio now incorporates two types of input data analysis that relate the ***Input Parameters*** of a model to the ***Responses*** defined in the experiment. Both of these are found in the Experiment window under the Input Analysis tab.

Response Sensitivity analysis is used to determine the sensitivity of the Responses to changes in the Input Parameters. Response Sensitivity analysis is very useful in situations where the input parameters have been specified in the absence of real world data and shows which Input Parameters have the biggest influence on the Reponses. Response Sensitivity analysis can be used early in a modeling project to help guide decisions on devoting time/money to initially collect real-world observations for input parameters.



In contrast ***Sample Size Error*** analysis is performed once some real-world data has been collected and is used to determine the overall impact of estimating the Input Parameters (based on real-world data) on the uncertainty in the Responses, to determine which estimated Input Parameters make the largest contribution to this uncertainty, and to identify the Input Parameters for which additional real-world observations would have the greatest benefit in reducing the uncertainty in the Responses.



For additional information, see the Help section on this topic as well as the newly added SimBit described below.

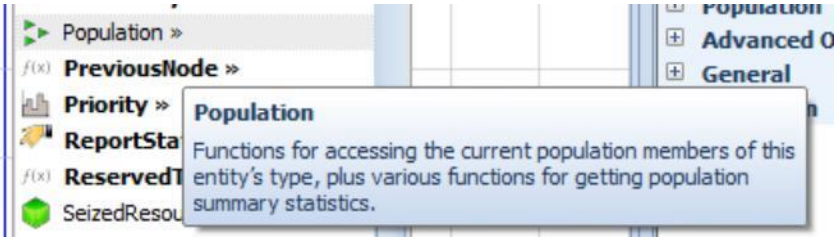
New SimBit

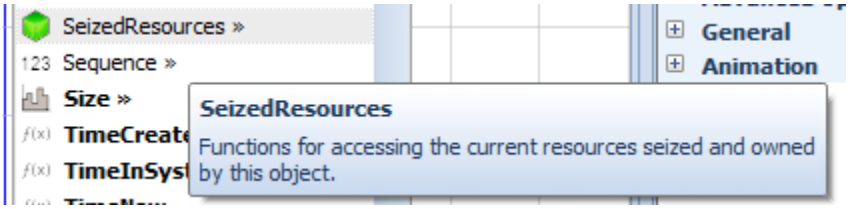
We've added a new SimBit to our SimBits library to support the above mentioned features:

InputAnalysis.spfx – This project consists of two models that demonstrate the new input sensitivity features within the Experiment window. Both the Response Sensitivity to input parameters, as well as the Sample Size Error based on input parameters, are included.

Enhanced Expression Editor (Auto-Completer)

As illustrated with several screenshot examples below, when you are using the expression editor, the drop-down auto-completion choices for all of the function namespaces now display icons as well as tool-tip description text.





Flow Enhancements – Suspend and Resume Steps

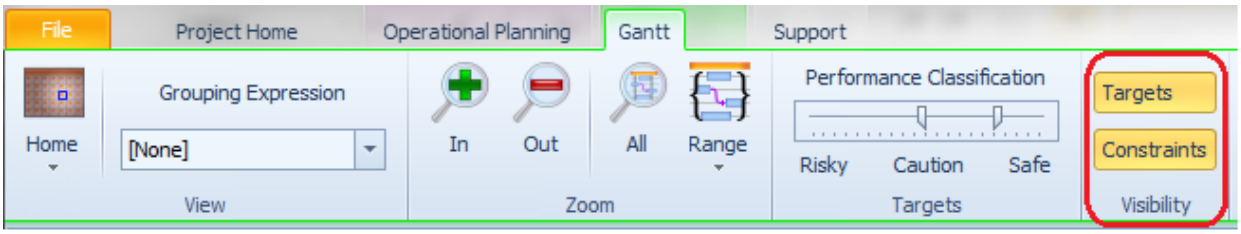
The Suspend and Resume steps have been enhanced to include the option to suspend and resume a flow regulator's flow capability.

This enhancement allows you to suspend/resume any flow through a regulator that is independent of assigning values to the regulator's Enabled state or CurrentMaximumFlowRate state, thus avoiding possible conflicts with other user-assignment logic. Additionally, because the Suspend step allows users to easily apply multiple simultaneous flow suspensions to a regulator, it will now also be easier to have multiple features independently suspending/resuming the flow capability of some specific regulator without worrying about possible conflicts between those features (e.g., one Tank feature renabling its 'Input' flow regulator when other active Tank feature still needs that regulator disabled).

The Tank & FlowToItemConverter objects in the Flow Library now use Suspend and Resume steps to suspend inflow into the object while the 'New Inflow Entering' add-on process is being executed (if that add-on process trigger is being used). We suspend the inflow if that add-on process is being executed to account for any possible customized 'Setup/Cleaning' delays that a user may choose to incorporate in that process logic when new inflow begins to enter the tank or the converter.

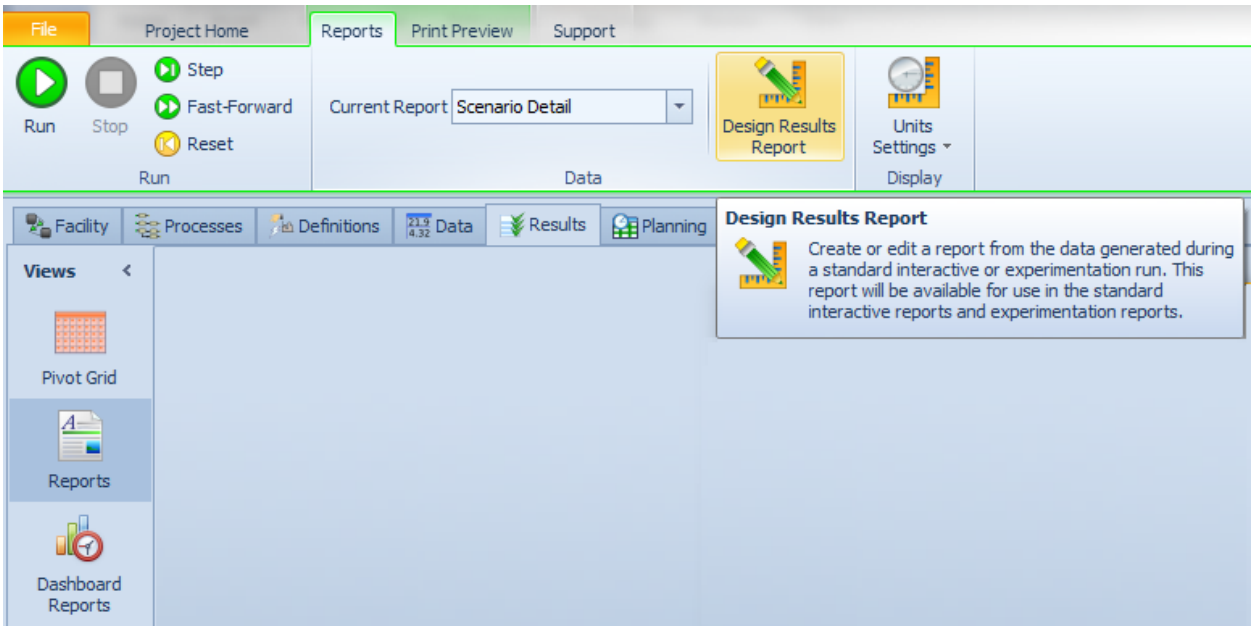
Simio Enterprise Edition - Visibility options on Entity Workflow Gantt

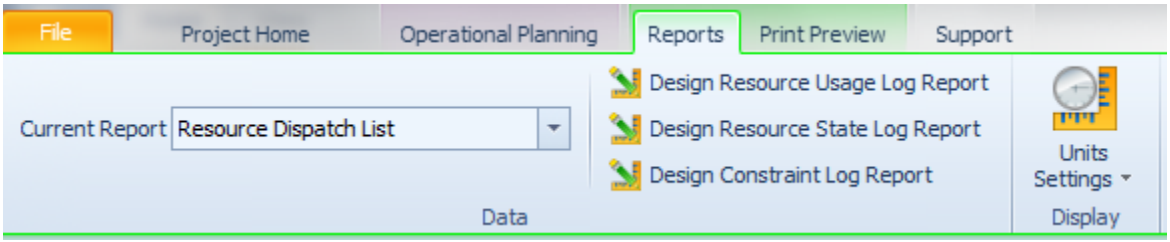
On the Gantt ribbon within the Entity Workflow window gantt, we've added buttons to toggle on and off the Targets and Constraints. Similar visibility options have been available within the Resource Plan gantt for several sprints.



Simio Enterprise Edition – Report Designer

The Report Designer allows users to create a custom Report based on generated model data. Reports can be made using data from the Interactive Model Results (i.e. the data displayed in the Pivot Grid), the Resource Usage Log, the Resource State Log and the Constraint Log. The Report Designer is available on the Reports ribbon.





API support for editing table schemas

The design time API now has the ability to add and/or delete columns and add or delete tables.

The IModel.Tables collection now supports: Create(), Create(name), Remove(table), RemoveAt(index), RemoveRange(index,count), and Clear(). The ITable.Columns now supports similar methods as above, as well as lots and Add[X]Column methods. See the API Reference Guide for more details.

Relational Table Performance Improvements

Setting table row references explicitly (SetRow step) or implicitly (Search step), should now be faster in the case of setting a row on table with either incoming or outgoing foreign key relations. This is expected to bring run performance improvement when using relational tables.

Settings GUI Option to Dock Windows

Within the File, Settings > Application Settings window, there is a new option under GUI to allow users to restore windows docking locations to the same location when Simio was last shut down. This was a user requested feature.

Repeat Group Window Sizing

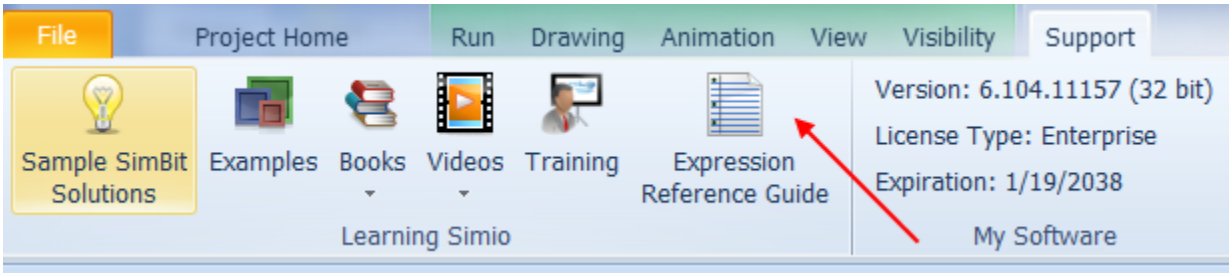
We now save the window size of the repeating editor, so if you resize it, then close it and open it again later (in the same Simio session or in a later one), it comes back in that same size. This was also a user requested feature.

Simio Release 6 – Sprint 104 – May 9, 2014

In Simio Sprint 104, we have added multiple user requests including the ability to open SimBits within the existing Simio instance, accessibility to changeover data with new functions and a new expression reference guide. Within the Processes window, we've added the ability to pass information in / out through the use of input arguments and return values. We've also made a few Experiment enhancements and Gantt changes based on user feedback.

Support Ribbon – New Expression Reference Guide

On the Support Ribbon, there is a new button that will generate an Expression Reference Guide custom created for the constructs used in your model.



Clicking on this button will generate an Expression Reference Guide html document which will be opened in your default browser. This document lists the Simio built-in functions, keywords, states and elements. It also lists the functions, states, events and elements for those objects that you have placed in your model. This can be very useful for finding and learning about the rich model information available in Simio.

Shift-Click to Open SimBit in Existing Simio Instance

In the past, within the .Sample SimBit Solutions window, a particular example was opened by clicking on the SimBit name. This would open the SimBit inside a separate instance of Simio using the unlicensed evaluation version of Simio. Within this sprint, you may now use the Shift key while clicking on the SimBit name to open the SimBit within the existing Simio instance. Any model that you currently have open will be closed and you will be asked to save the model before closing if changes had been made.

Access to Changeover Matric Data in Expressions

The ability to access changeover matrix data in expressions is now supported. A changeover matrix value may now be referenced in any expression using the syntax **MatrixName**[*row*,*column*], where the *row* and *column* subscripts are numeric constants or expressions that return zero-based row and column indexes into the matrix. When you add a changeover matrix to a model, you will see the available changeover matrix syntax within the expression builder.

Also implemented is the ability to support specifying the name of a changeover matrix in an object property (repeating or non-repeating). Then that object's process logic is able to get to the matrix data through the property reference. First you do this by adding a Changeover Matrix Property to the model by going to Definitions/Properties and selecting Standard Property and Changeover Matrix. Then in the

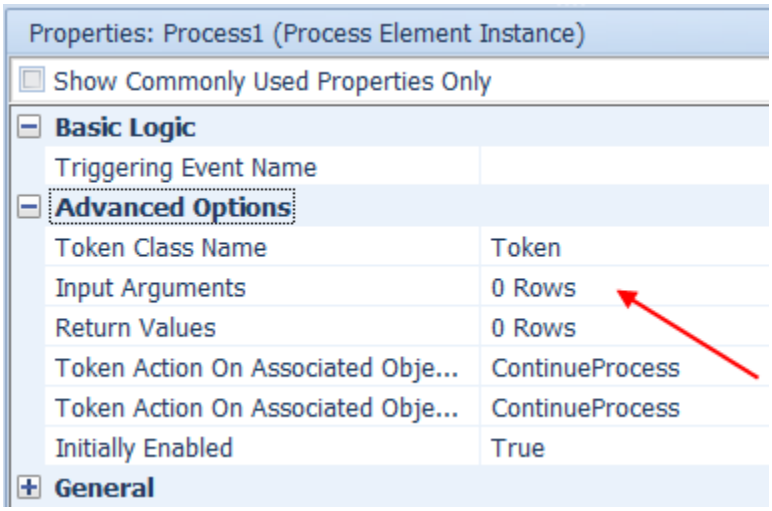
process logic of the object that has the property, you can use the syntax for ***ChangeOverMatrixPropertyName.Matrix***[*row,column*] or ***RepeatGroupPropertyName. ChangeOverMatrixPropertyName.Matrix[****row,column*] to access a value at a specified row and column index location in the changeover matrix that is being referenced by the property.

Experiments – Freeze Scenario Name when Scrolling Right

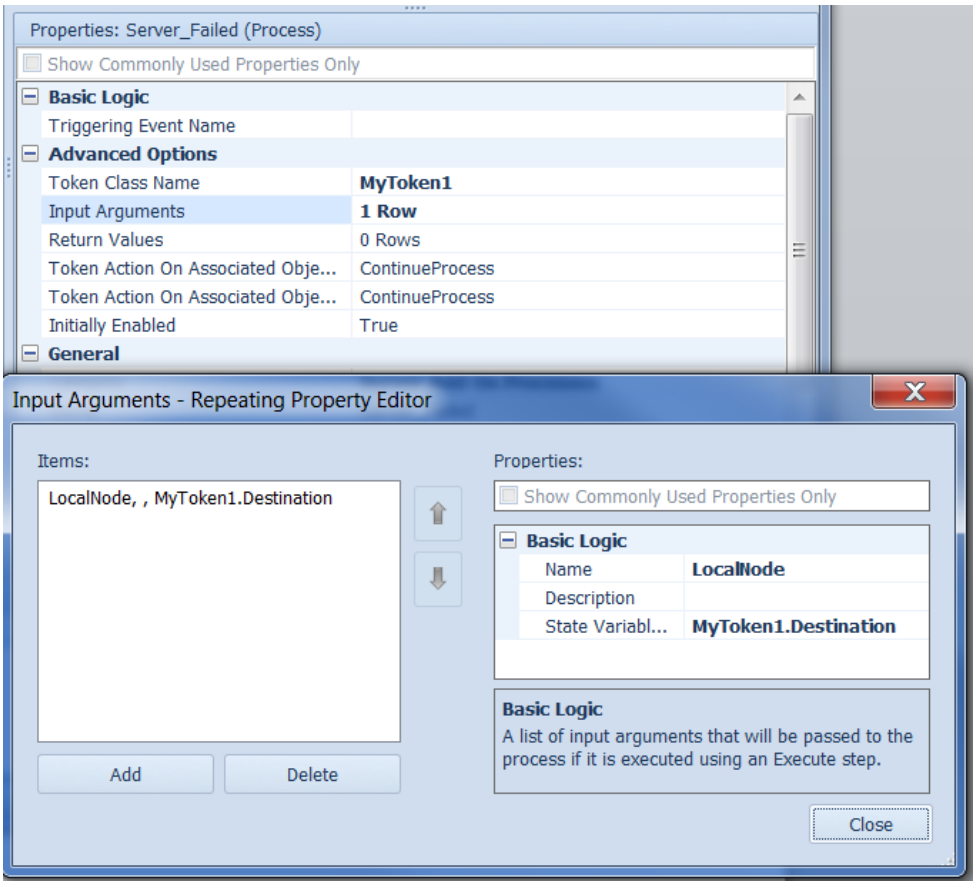
In the Design tab of the Experiment window, the Scenario Name and Scenario Status columns now remain stationary when scrolling to the right in the window.

Multiple Passed and Returned Values on Execute Step

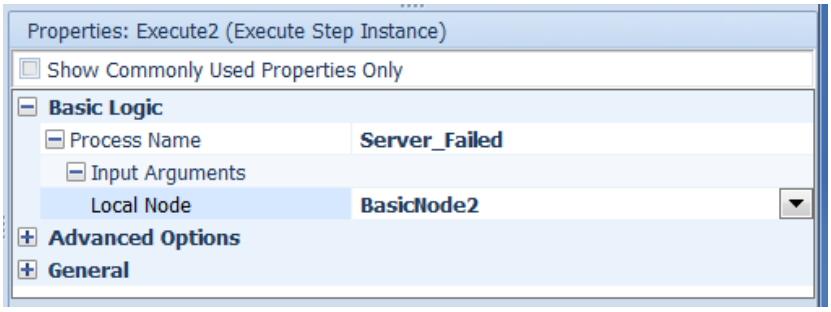
Simio now supports the ability to pass input arguments and return values on executed processes. This allows single shared processes to be executed within multiple objects. Processes now have a repeat group where you can identify, for the token class, the *Input Arguments* and *Return Values.*



For tokens *Input Arguments*, the State Name Variable should always be a state on a custom token, where that custom token is specified as the Token Class Name of the parent process. Doing otherwise could cause modeling problems. Making it a token state makes sure that each token has its own unique value. Specifying a state variable here indicates that when the process is executed, the value of this Input Parameter will be assigned to that state.



Once the *Input Arguments* have been defined, they will appear anywhere their parent process is referenced (ex: in an Execute step, an Add-On Process, etc.)



Note that the Input Arguments will pick up the unit type (if any) from the state referenced in the State Variable Name of the *Input Argument*. Refer to SimBit MultipleInputArgumentsOnProcess for an example on how to use this feature.

For token *Return Values*, the expression here is evaluated in the context of the process and it's token. When the process is finished, the value of this expression is assigned back to a state variable specified. These Return Values will appear anywhere their parent process is referenced (ex: in an Execute step or an Add-On process).

New SimBit

We've added a new SimBit to our SimBits library to illustrate the above described feature:

> **MultipleInputArgumentsOnProcesses.spfx** – This model shows how the input arguments on a process can be used to store information for multiple objects accessing the same process within a model.

Unique Experiment Run ID

Simio now generates unique identifiers for each experiment run. These identifiers include **Run.ResultSetID** and **Run.ResultSetTimeStampUtcString**. These unique identifiers are useful when writing data externally from Simio. It supports the unique identification of which experiment run the data was generated on. The **Run.ResultSetID** returns a unique identifier for the results produced by a run. Separate identifiers exist for interactive, plan or experiment runs, each of which is assigned a new value whenever the corresponding modle, plan or experiment is reset. The **Run.ResultSetTimeStampUtcString** returns the DateTime that the current ResultSetID was created. This is not a simulation date and time. It is the actual date and time, expressed as Coordinated Universal Time (UTC).

Simio API Enhancements

Simio now provides a way to view the additional column data on the Resource Usage Log through the API. Also added is the ability to access the Material Log, State Log and Tally Log data through the API. Please refer to the Simio API Reference Guide for more details.

Enterprise Edition - Gantt Enhancements

User Defined List State can now be displayed in the Resource Plan Gantt. If an object that is Logging Resource Usage and therefore appearing in the Resource Plan Gantt, and it contains a State Statistic for a user defined List State, the State Statistic will be displayed in the Resource Gantt if the *Log Observation* property on the State Statistic is set to 'True'. There is a new button in the Gantt Ribbon that controls if State Statistic rows should be displayed or hidden.

Resources and entities that have not completed a Seize at the end of the model run will now be displayed in the Gantt charts, with three dots following at the end. This indicates that the End Time of the Seize is unknown and has not happened as of the time of the run completion. Previously, an entity or a resource that had not been released when the run ended would not appear on the Gantt.

Simio Release 6 – Sprint 103 – April 15, 2014

Within the past several sprints, we have continued to enhance our Flow Library with new features. This has continued into this sprint as well, with an update to the converter objects and addition of automatic fill status statistics on objects with containers. In this sprint, we've also made several user requested changes, including additional options on status pie charts, API support for symbol changes and new documentation features for processes and steps.

Flow Library Enhancements – Stop Early Event Name for Converter Objects

We have extended one of theFlow Library enhancements from Sprint 102 (Stop Early Event Name) to the ItemToFlowConverter and FlowToItemConverter objects. These objects now have an optional event to allow additional control over triggering the stoppage of a conversion process early before reaching the specified 'Flow Quantity Per Item' target. This feature was put into the Filler and Emptier in the previous sprint.

Flow Library Enhancements - Container Statistics on Tank, Converter Objects and ContainerEntity.

For the objects with associated flow containers, we've added statistics on the state of the flow container. This includes the Tank, ItemToFlowConverter, FlowToItemConverter and ContainerEntity objects. The statistics include such information as TimeEmpty, TimeFull,and TimePartiallyFull. Tank also includes VolumeCapacity or WeightCapacity.

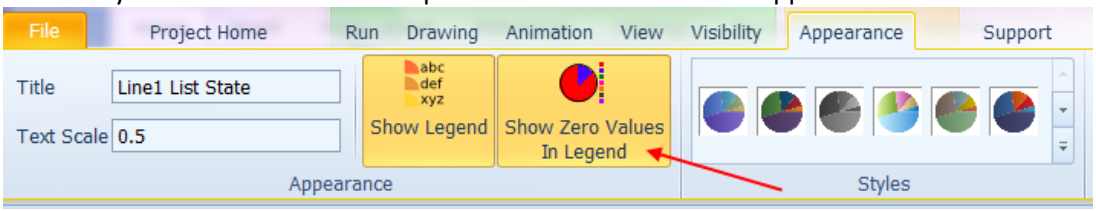
These statistics were possible because of a new Container AutoStates associated with the Container element. These built-in autostates include Empty (0), PartiallyFull (1) and Full (2). The objects within the Flow library then include a 'FillStatus' list state variable based on these autostates.

Object Type	Object Name	Data Source	Category	Data Item	Statistic	Average Total
Tank	Tank1	FlowContainer	Capacity	TimeEmpty	Average (Hours)	23.7900
					Occurrences	1.0000
					Percent	99.1250
					Total (Hours)	23.7900
				TimeFull	Average (Hours)	0.0600
					Occurrences	1.0000
					Percent	0.2500
					Total (Hours)	0.0600
				TimePartiallyFull	Average (Hours)	0.1500
					Occurrences	1.0000
					Percent	0.6250
					Total (Hours)	0.1500
				VolumeCapacity	Average (Cubic Meters)	15.0000
					FinalValue (Cubic Meters)	15.0000
					Maximum (Cubic Meters)	15.0000
					Minimum (Cubic Meters)	15.0000
			Content	VolumeLevel	Average (Cubic Meters)	0.0919
					FinalValue (Cubic Meters)	0.0000
					Maximum (Cubic Meters)	21.0000
					Minimum (Cubic Meters)	0.0000

Status Pie Enhancements - Legend Labels with Spaces and Option to Show Zero Values

When displaying a Status Pie legend with List states that have no spaces, spaces are now automatically entered when there is a capital letter in the name. For example, a List value of *Line1Idle* will appear in the pie chart legend as *Line1 Idle*.

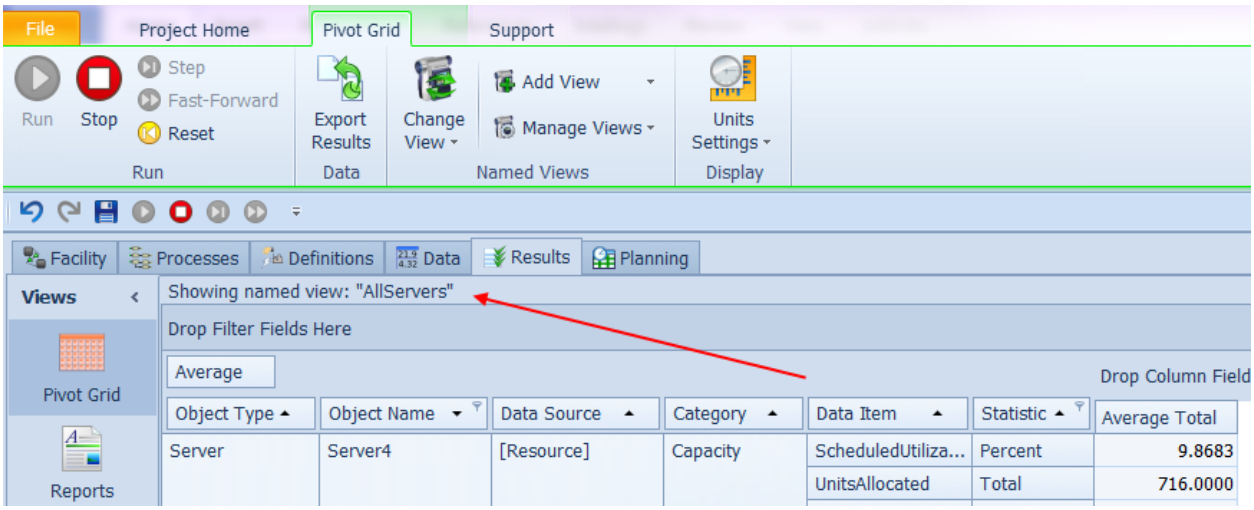
The ability to hide zero values on a pie chart is available on the Appearance ribbon.



Toggling this button on or off will enable or disable zero values of the Repeat Group or List expressions in the pie chart legend.

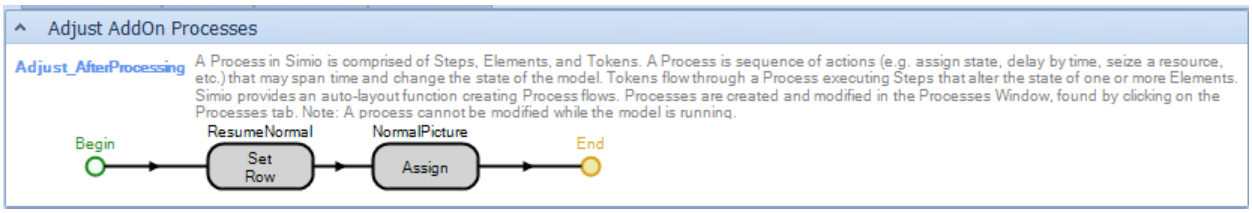
Displaying the Named View on Pivot Grids

When naming a view on the pivot grids, the name of the view now appears under the pivot grid tab. This is true for pivot grids generated during a model run or during experimentation.

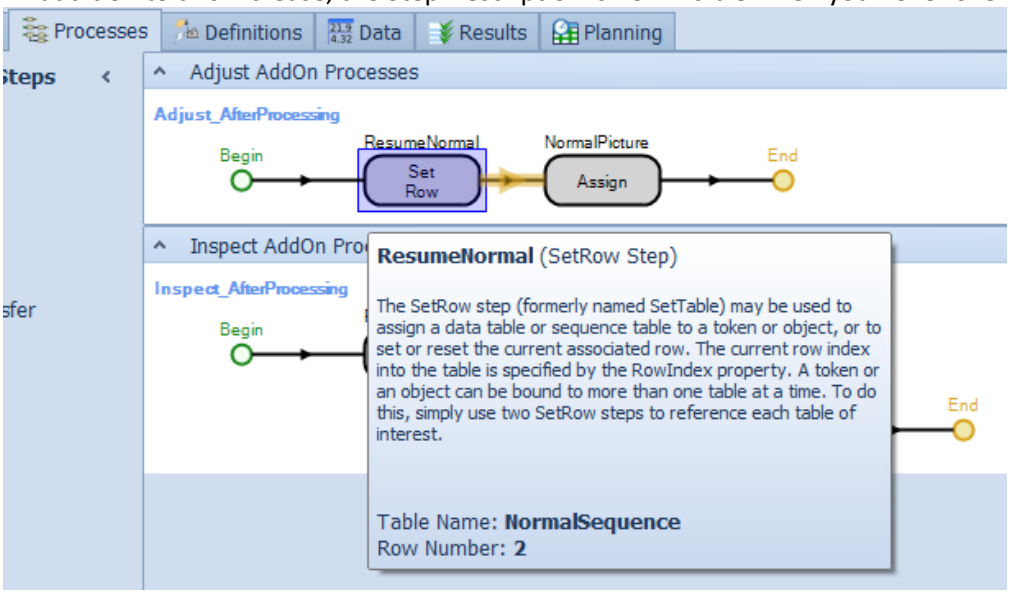


Process and Step Description Changes

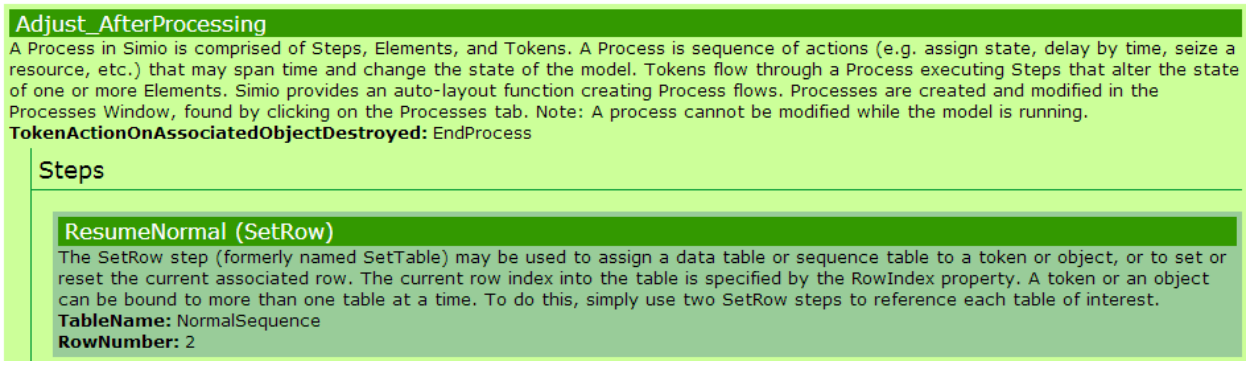
There have been modifications made to the Process window to allow for better Process and Step documentation. The first is an increase in the length of a Step's Description property.



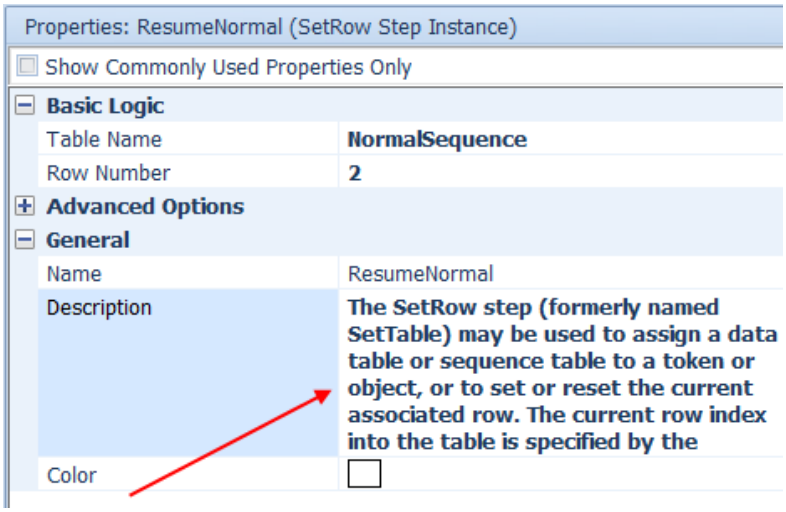
In addition to this increase, the Step Description is now visible when you hover over the Step.



The Step Description now appears in the Documentation Report.



Finally, the Process Description property has been increased from the previously displayed 2 lines.

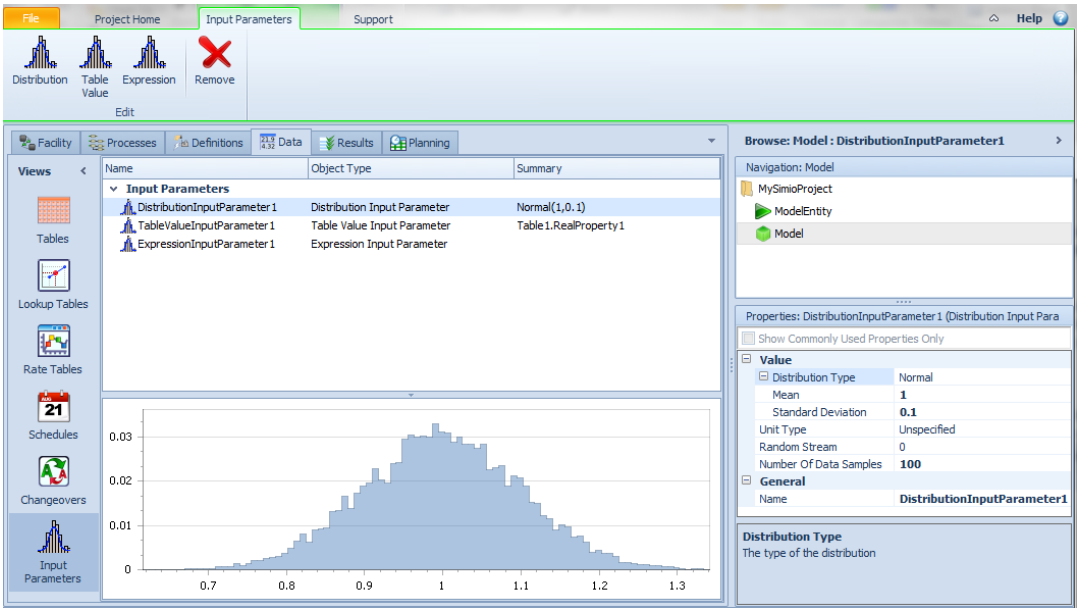


List State Enhancements

The Results Classification of properties, including Data Source, Category and Data Item, have been added to the List State type of State variable within the Definitions / States of a model. This feature provides user-control over specifying the various properties used to classify results automatically reported by a list state variable.

Input Parameters – New Distributions and Table Value

The new Input Parameters section of data within the Data tab now includes two additional input parameter types, including Distributions and Table Values. With distribution type, the input parameter defines data that can be sampled during the model run. This typically is taken from real world data that has been sampled and a distribution fit. With table values type, the input parameters define data that is sampled directly from a table column. These input parameters can then be evaluated within the Input Sensitivity section of an Experiment to determine how sensitive various responses are to those inputs.



API Support for Getting/Setting Symbol Index

Note there is a new API support dll named SimioAPI.Graphics.dll that must be used alongside SimioAPI.Extensions.dll.

In a design add-in, given an IDesignContext called context, you can do something like:

```
var theobject = context.ActiveModel.Facility.IntelligentObjects[0];
var graphic = context.ActiveModelGraphics.GetSymbolGraphicsFor(theobject);
graphic.CurrentSymbolIndex = 5;
```

Simio Release 6 – Sprint 102 – March 25, 2014

We have continued to enhance the Flow Library objects by adding a Stop Filling/Emptying event on Filler and Emptier, as well as adding purge content triggers to those objects with container elements. We've also added some Experiment based enhancements, such as allowing a scenario's control values to easily be fed into a model, as well as a new Input Sensitivity tab (and supporting Input Parameters within the model) for analysis. Of course, we continue to take into account user requests and have added color coding to steps within the Processes window.

Enhanced Subscribe / Unsubscribe Steps

We have made changes to the Subscribe and Unsubscribe steps to support some of our Flow Library enhancements. These enhancements additionally make these steps more useful as well. Within the Subscribe step, a new property, *Event Condition*, provides an optional condition to be evaluated whenever the triggering event occurs. This condition must also be true to cause execution of the specified process. Within both the Subscribe and Unsubscribe steps, the Event (More) repeating properties have been added to allow for multiple events and processes to be specified within a single step.

Filler and Emptier Enhancement – Stop Filling / Emptying Event

Both the Filler and Emptier now allow a user to optionally define the name of an event that will stop filling or emptying a container early, before reaching the specified fill or empty target.

A simple example for a Filler might be to fill a container until it is full or until a Source tank is empty (the tank becoming empty being the 'stop early' event), whichever occurs first. Similarly, an Emptier might be emptying a container until empty, or the destination Tank is full, whichever occurs first. This new property can be used to model any kind of 'Stop Early' signal situation.

Tank, ItemToFlowConverter and FlowToItemConverter Enhancements - Purge Content Triggers Repeating Properties

The Flow Library Tank object, as well as the ItemToFlowConverter and FlowToItemConverter objects, now provide some optional event-driven triggers (using a basic 'event-condition-action' rule design), that will immediately remove and dispose all of the Tank's contents, putting the Tank into an empty state. Multiple *Trigger Event Name* properties and associated *Condition* properties can be specified for each object to immediately empty any contents in the tank or converter's flow container within the associated object. This is useful when modeling different product types and cleaning the tanks upon changeovers and such.

ObjectType.List.ListName.ListEntry Syntax in Expressions

You can now reference the List keyword inside of an object to get a reference to a value from a List. For example, to assign a value from a String List located in the ModelEntity to a List State located in the ModelEntity, you can now use the syntax: ModelEntity.ListState == ModelEntity.List.StringList.Value. The SimBit named Select Entity Color from String List has been updated to use this new syntax. * Note: In the

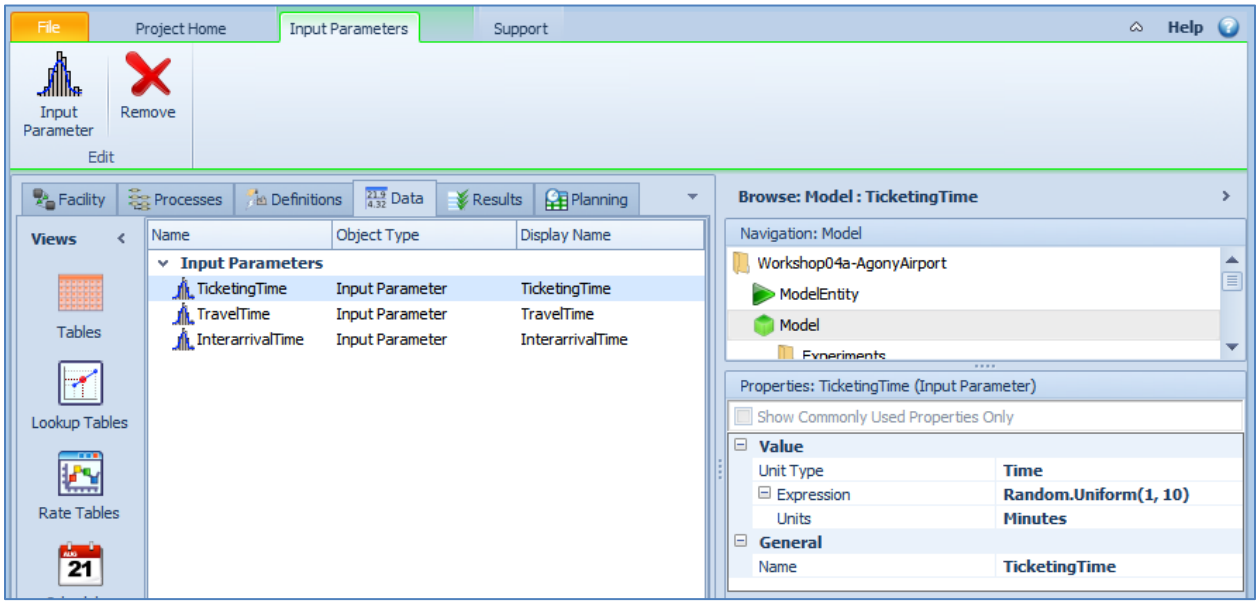
past versions of Simio, to access that list for an object such as ModelEntity, you needed to have a duplicate list within the Model; this is no longer necessary.

Experiment Scenario Control Values Can Now Be Copied to Interactive Model Properties

As a user requested enhancement, there is now a right-click option on the Scenario column within an experiment. Right-clicking on a particular scenario will now allow users to 'Copy this scenario's control values to the model.' This option will take the values of all the Controls specified for that given scenario and push them into the interactive Model associated with that Experiment.

New Input Parameters (Model's Data Tab) and Input Sensitivity (Experiments)

Within the Data tab of a model, there is now a new Input Parameters panel icon. An Input Parameter allows you to define an expression. The input parameter name can then be used in another expression within the model. For example, you could define an Input Parameter with the *Name* 'TicketingTime' with the *Expression* value 'Random.Uniform(1,10)', as shown below. Then, in a Server's *Processing Time* property, you may reference that input parameter as 'TicketingTime + 5'.



Right-clicking on a property value of a given object also allows you to 'Set Input Parameter' or 'Create New Input Parameter', similar to setting a referenced property value.

In conjunction with the Input Parameters, we've added the ability to run sensitivity analysis on the various input parameters defined from within the Experiment. Within a given experiment, there is now an Input Sensitivity tab that includes the Bar Chart, Pie Chart and Raw Data window ribbons.

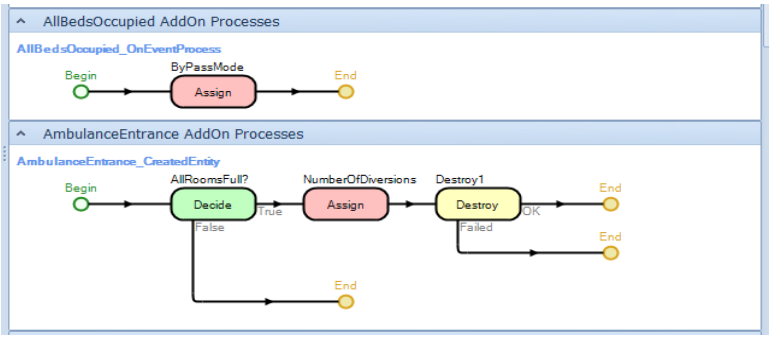
This collection of features represents our latest contribution to make it easier to analyse your data and to determine how reliable it is and where you might implement improvements. For example the diagram above indicates that the Input Parameter named *TicketingTime* accounts for about 22% of the tracked variability in *TimeInSystem* and 100% of the variability in *AgentUtil*. This provides objective information to help judge the merits of improving the data or estimates for the various inputs.

Enhanced Link Arrows

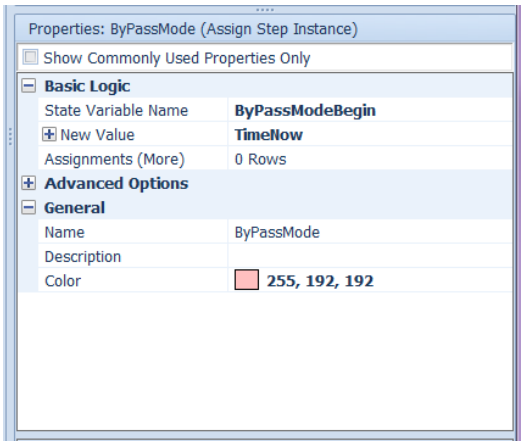
We have reduced the size of the directional arrows on our links so that they are visible when smaller sized links are placed. The location and performance of the arrows has also been improved.

Processes Window Enhancement – Allow Colors on Steps

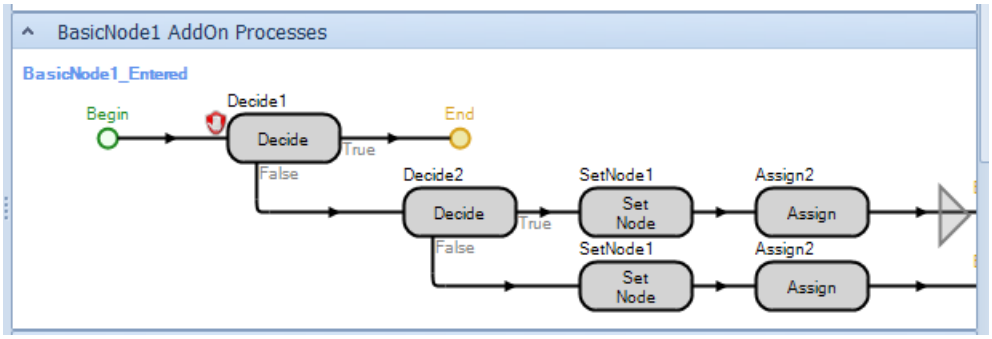
Within the Processes Window, there is now an option to color steps in a process, like in the example shown below. This had been requested by a number of users to help distinguish steps or sections of processes from other sections.



The step color is a property that's available under the **General** heading.



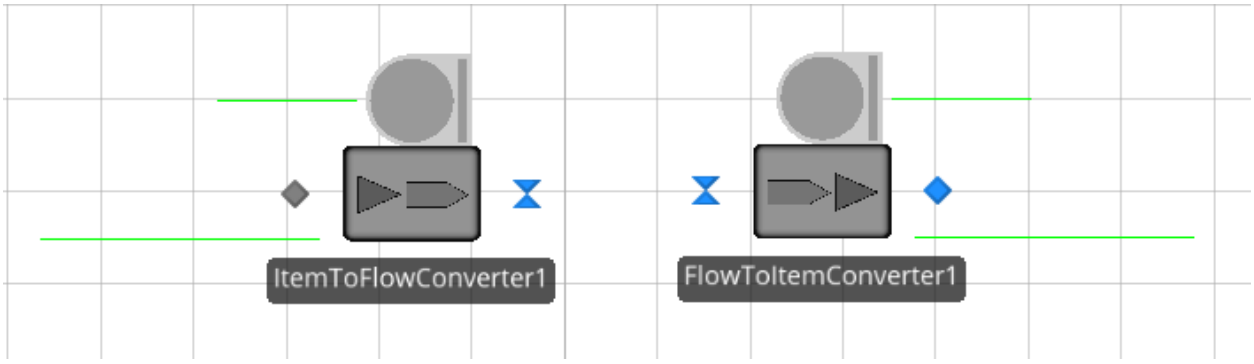
In previous versions of the software, setting a breakpoint on a step in the Processes window changed the color of that step to red. With the ability to add colors to steps, we've changed how breakpoints on steps are identified. They will now have the breakpoint symbol to the left of the step, similar to how breaks are identified in the Facility window.



Simio Release 6 – Sprint 101 – March 4, 2014

We have added two frequently requested features in this Simio sprint, as well as made several new objects and enhancements to the Flow Library. Our new Properties Spreadsheet window allows users to conveniently view and edit properties of all objects of a selected type within a spreadsheet type format. We've also modified the way that && and || operators are evaluated within complex expressions, enabling the 'short circuit' of those expressions when the first portion of the expression evaluates to true/false.

New Flow Library Objects – ItemToFlowConverter and FlowToItemConverter



The ItemToFlowConverter object may be used to model a process that converts entities representing discrete items into flow entities representing quantities of fluids or other mass. The flow rate out of the object is regulated by the 'Output' flow node. Discrete item entities enter the object through the 'Input' basic node and are then processed into flow.

The FlowToItemConverter object may be used to model a process that converts flow entities representing quantities of fluids or other mass into entities representing discrete items. The flow rate into the object is regulated by its 'Input' flow node. Discrete item entities exit the object from its 'Output' transfer node.

Flow Library Enhancements – Tank

We have added some user requested enhancements to the Tank object within the Flow Library. Changes include the addition of the new level locations of 'Low Low' or 'High High' marks, as well as optional add-on process triggers and built-in state assignments for when the tank level is rising above or falling below those marks.

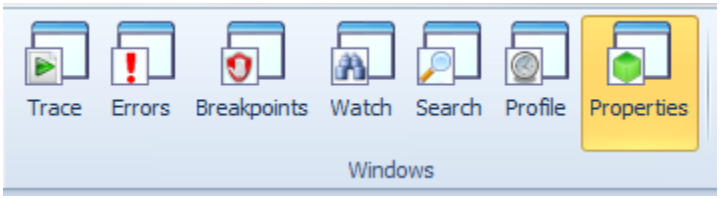
Additionally, within the Tank object, we've added 'New Inflow Entering' add-on process trigger and state assignment features. These optional state assignments and add-on process triggers are used when a new inflow is starting to enter the Tank object, which may be either the first inflow into an empty tank or a change to the inflow entity type into a partially full tank. This functionality is consistent with the new FlowToItemConverter object as well.

New SimBit

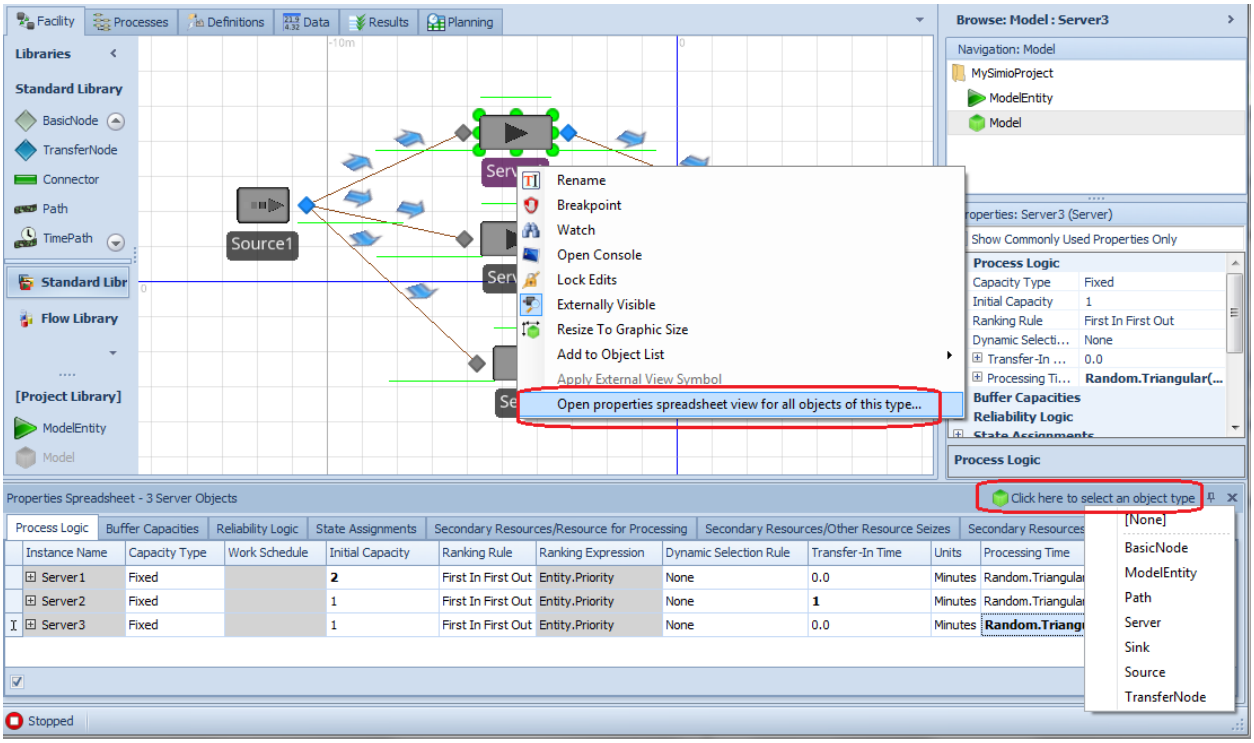
We've added a new SimBit to our SimBits library:

InfectionPropagationUsingContinuousAndFlow.spfx – This project includes two models demonstrating two different ways to model a continuous system / system dynamics with an infection propagation example. The UsingFlowObjects model utilizes the Flow Library objects including Tanks and FlowConnectors, while the UsingLevelStates model utilizes state continuous variables and functions.

New Properties Spreadsheet Window



Within the Project Home ribbon, Windows section, we've added a Properties button that will open the new Properties Spreadsheet window, as shown below. This window allows property data for specific objects to be viewed and edited within a spreadsheet format. To view data for a particular object (i.e., Server, Source, Path, etc.), either select the object type from the top right of the Properties Spreadsheet window ('Click here to select an object type' and select the type of object) OR right click on an object of that type and select 'Open properties spreadsheet view for all objects of this type… '.



Within the Properties Spreadsheet window, you can edit single property entries for a given object or copy/paste values into multiple columns. To select multiple entry rows, you may click a row, then use

Ctrl-click to highlight additional rows – editing of multiple rows simultaneously can continue to be done within the Properties window on the right.

Filtering and sorting the object list can also be done based on a particular column, similar to table functionality. Scroll bars are also available for objects with many properties and/or categories. Right-clicking on a particular row will allow you to 'Navigate to ' the particular object selected within the Facility window.

'Short Circuit' Evaluation of && and || operators

We now 'short circuit' evaluation of the && and || operators. What this means is that if the first part of expression including the && operator is False, then the entire && operation returns False, and we don't evaluate the second part. Similarly, if the first part of an expression with the || operator is True, then the entire || operation is True, and we don't evaluate the second part.

This means that saying something like Math.If(a && b, 1, 2) Is now equivalent to what you could already do with nested Math.If() statements, Math.If(a, Math.If(b, 1, 2), 2). Similarly, something like Math.If(a || b, 1, 2) is now equivalent to Math.If(a, 1, Math.If(b, 1, 2)).

This 'short circuit' evaluation is very useful when you have references that are potentially set to nothing but you want to do evaluations with them, so instead of nested Math.If() statements you could simply write Math.If(MyRefState != Nothing && MyRefState.MyEntity.MyState == 5, 100, 200)

In older software, if you had this, and we evaluated it when MyRefState was Nothing, we would try to evaluate both parts, leading to a runtime error when trying to cast a Nothing to MyEntity. Now however, it would work fine.

Note that there is a new advanced compatibility setting "Ignore Short Circuit Evaluation In Expressions", which is True for old models. Setting it to True makes us evaluate in the 'old' way (evaluating all parts of the && or || no matter what). We have it because things like taking random samples in the expression has a side effect, and we want to make sure that old models have the same result. It is set to False (use the new behavior) for new models.

Per Day and Per Week Options for Volume and Weight Rates

For Properties, States and Expressions that have a *Unit Type* property value set to 'VolumeFlowRate' or 'WeightFlowRate', we've now added various units per day and units per week options to the list of *Default Units*.

Design-Time Add-Ins Can Specify Where They Appear in the Ribbons

Users who write their own design-time add-ins can now specify where their add-in appears within Simio's ribbons. There is a new optional interface that can be implemented by the add-in to specify the ribbon category, ribbon tab, and/or group where the add-in should be shown. See **IDesignAddInGuiDetails** in the *Simio API Reference Guide* for more information.

There is a sample implementation of this functionality in the C:\Users\Public\Documents\Simio\Examples\UserExtensions\SourceServerSink add-in example.

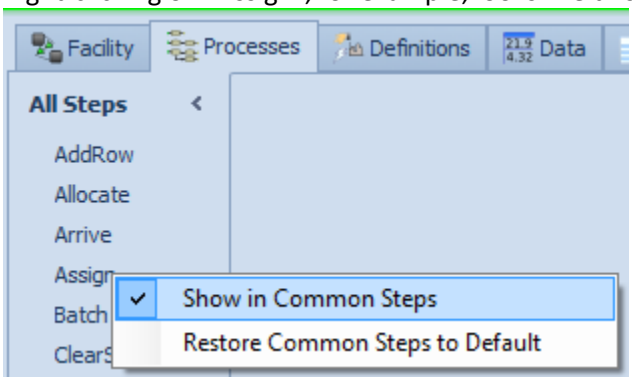
Simio Release 6 – Sprint 100 – February 13, 2014

In this Simio sprint, you will find a number of user requested changes, from customizable Common Steps to design time add-in enhancements to graphics / animation features. We've also added several SimBits to our extensive library of small examples.

Customizable "Common Steps" Panel

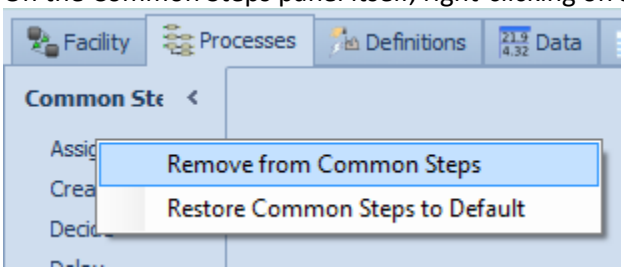
You can now choose your own Common Steps. If you're looking at a *different* steps panel (All Steps or User Defined), you can right-click on a step and add or remove it from Common Steps.

Right-clicking on "Assign", for example, looks like this:



Note that the first menu item is checked, meaning that it is currently shown on the Common Steps panel.

On the Common Steps panel itself, right-clicking on step allows you to remove it:



You can also use drag and drop to re-arrange the steps on the Common Steps panel. There's also the option of resetting to Simio's default set of steps.

This is a product-level feature (not per-project or per-model) so changes in one Processes window propagate to all others.

You now also may adjust the width of the Step panels.

New Graphics Type in Settings

There is a new Graphics Type option in the Settings called 'SoftwareDirectX' which is equivalent to using the command line -force-software-mode switch. This is useful for customers running on a VM where Simio thinks it can use DirectX but really cannot.

Changed Default Values when Creating New Referenced Property

We have changed the behavior slightly for the default values of model properties that are defined using the right-click on property > Set Referenced Property > Create New Referenced Property. In previous software sprints, if you right-clicked on a property of an object that you wanted to change to a model property (for experimentation), the default value of that new model property would be the default value of the property from which it was defined. For example, if you chose to Create New Referenced Property from the Processing Time property of a Server, the default value of the new model property would be 'Random.Triangular(.1,.2,.3)', regardless of whether that was the current value for that Server's processing time property.

The software now will take the current value of the object property from which the model property is created and that will be the default value for the model property. Thus, if you chose to Create New Referenced Property from the Processing Time of a Server that has a value of 2.5, the default value of the new model property will be '2.5'.

Create Step - New 'CopySpecificObject' Create Type

The Create step has been enhanced to include a new 'CopySpecificObject' option on the *Create Type* property. An additional new property, *Source Entity Object*, has also been added to correspond with the copying of a specific object to indicate the specific entity object from which to copy. The Object Instance Name property has been renamed to *Entity Type*.

New SimBits

We've added three new SimBits to our SimBits library, including:

- **SimpleElevatorObject.spfx** – This project contains a simple elevator object. It consists of 2 transfer nodes and a bi-directional path connecting the two nodes. A standard vehicle object is used as the elevator platform that travels between the two nodes. There are two models in this SimBit that use this elevator object to transport entities between floors.

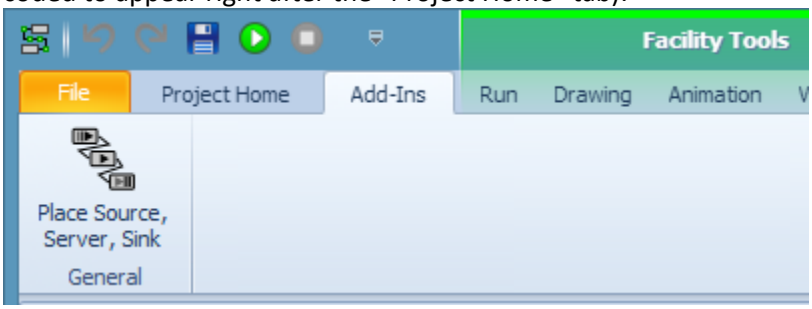
- **AnimatePathDensity.spfx** - This model demonstrates one approach to animating the travel density of a path. This is done by changing the width of each path so that the width is equal to the percentage of travelers that have traveled this path.

- **RandomValueFromTable.spfx** – This model illustrates the new 'RandomValue' function for generating randomly selected data from a table (added in previous sprint). The top subsystem illustrates using data where you have a standalone set of collected values, for example historical processing times. The bottom subsystem illustrates using data where you have a set of collected values with identifying or categorizing information, for example historical processing times by entity type.

Design-Time Add-Ins Can Specify Where They Appear in the Ribbons

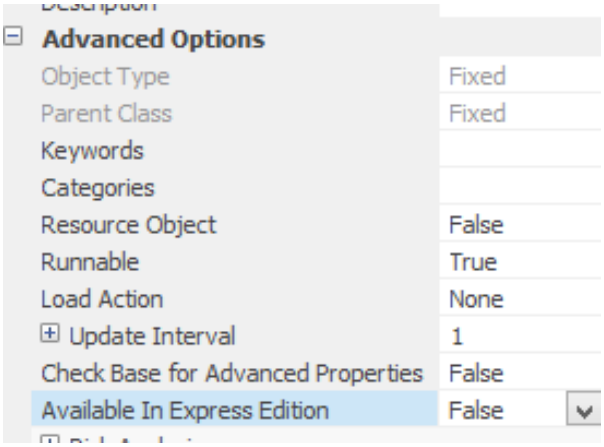
There is now an optional interface (IDesignAddInGuiDetails) that an add-in can implement to allow it to show up in the ribbon. You can specify CategoryName, TabName, and GroupName to have full control over exactly where your add-in button will appear.

All three of these properties are optional. If you simply implement the interface and then return null for each property, the add-in will appear in a new "Add-Ins" tab, in a group called "General" (currently coded to appear right after the "Project Home" tab):



If you return a different value for GroupName, we'll use that instead of "General". Similarly, if you return a different value for TabName, we'll create another tab with that name. Finally, if you return a value for CategoryName, we'll place the tab under a new category (currently to appear right before the "Support" category).

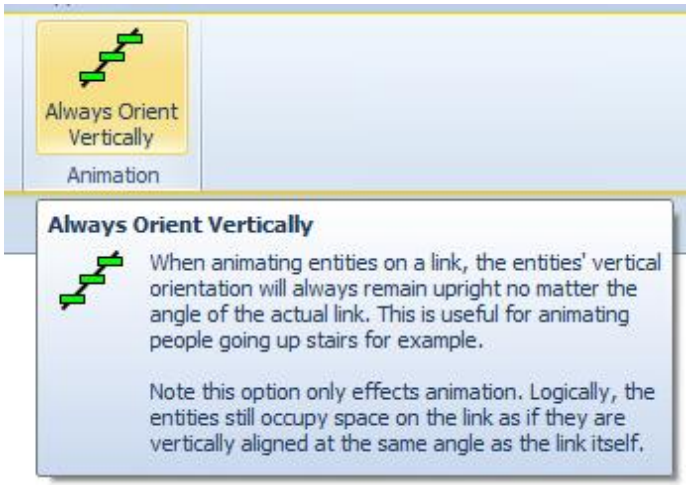
New Express Edition Openness



We have provided users of Enterprise Edition the ability to specify that a particular object may be loaded into Express Edition. Previously Simio Express was restricted to loading only the Simio Standard Library.
Corporate groups and independent consultants wishing to deploy a custom Express Edition-based solution will find this particularly useful.

Always Orient Vertically option for Link Animation

We have added an option to the Edit ribbon when a Link is selected in the Facility window. This option allows for entities to have an upright orientation regardless of the link angle. This is particularly useful when your link is changing height (like an escalator or inclined Power & Free conveyor) and you want the entities to remain upright, rather than traveling at an inclined angle.



Simio Release 6 – Sprint 98/99 – January 20, 2014

Happy New Year! With this New Year brings many new user requested features, ranging from selecting random values from a table column to material related statistics to a time in system logging option on the Sink and many more.

New Table Function for Selecting Random Value

Many times users have simulation data in the form of real actual data which then must be read into an input process / analyser to determine the best fit distribution to use within the simulation model. We have added a new table function that will allow users to alternatively input the actual data into a table and randomly select the data from that table.

TableName.PropertyName.RandomValue - returns the value of the property (table column) for a randomly selected row in the table, where each row has an equal probability of being selected.

If the table has primary key/foreign key relationships, and the active token or object is referencing a row selection set in the table, then this function will return the value for a row randomly selected from the related rows only. Otherwise, the row will be randomly selected from all rows in the table.

If randomness is disabled in a model and the 'RandomValue' function is utilized, the following behavior will be exhibited. If the table property (i.e., table column) is a numeric property, then the 'RandomValue' function will return the average value of all rows for the property. If the table property is non-numeric (e.g., a column of strings or object references), then the function will return the value of the middle row in the table (i.e., the row value corresponding to .5 cumulative probability).

Table RunTime Performance Improvements

We have made some runtime performance improvement changes for table selections, especially with tables that have a lot of relations between them. Some models with table relations may run up to 3 times faster when making selections on the master table. Models with no table relations should see some performance improvements as well.

Simplified Expression Editor to Exclude Function Optional Arguments

We have simplified the use of expression editor to make it easier to enter the most common expressions. The optional parameters are still noted in the pop-up tips and may still be used in the expression, but no longer must be deleted when you are not using them. For example, when using a random uniform distribution, the expression editor will now list Random.Uniform(min, max) and not include the [stream].

Time In System Logging Option on Sink

We have added the option on the Sink object within the Standard Library to *Log Time In System Observations*. By specifying this property as 'True', the individual observations through the Sink will be logged to the Tally Observation Log in the Logs panel of the Results tab.

Material Element Functions / Statistics

We have added three new functions to the Material element and statistics on these functions are automatically generated:

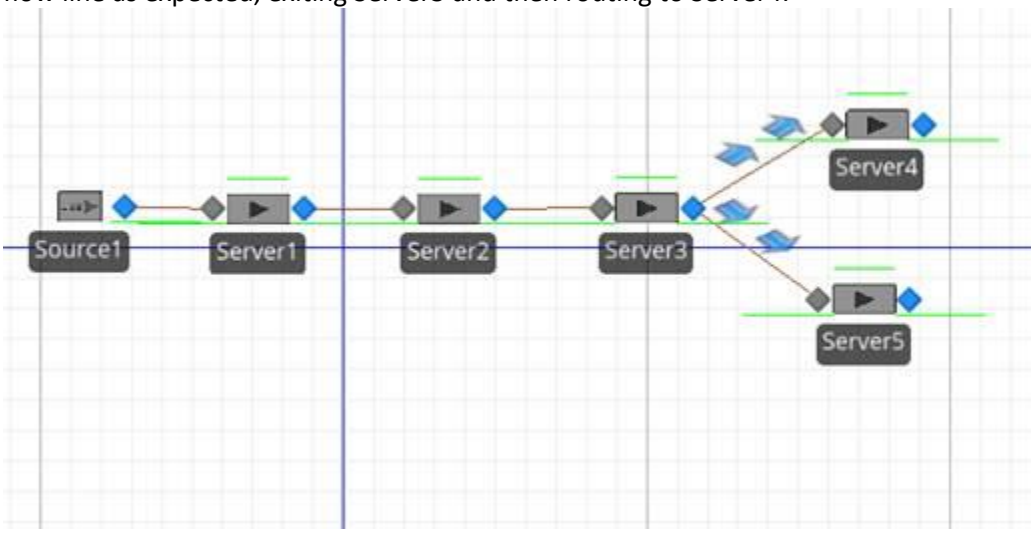
AverageQuantityAvailable- Returns the average quantity of this material that has been available in the system during the run.
MinimumQuantityAvailable- Returns the minimum quantity of this material that has been available in the system during the run.
MaximumQuantityAvailable- Returns the maximum quantity of this material that has been available in the system during the run.

Pickup and Dropoff Steps Enhancement

The Pickup and Dropoff steps now include a *Node Name* property so that a transporter is no longer required to be located at the node where a pickup or dropoff action is occurring.

Change in Behavior when Entity Enters an 'Input' Node

When an entity enters the 'Input' node for entering an attached associated object, if the input node has no drawn outbound links then the engine will now always assume the entity's objective is to enter the associated object regardless of its current assigned destination. For example, in the model below, if you assigned the destination of an entity leaving 'Source1' as 'Server4', the entity will now flow through the flow line as expected, exiting Server3 and then routing to Server4.

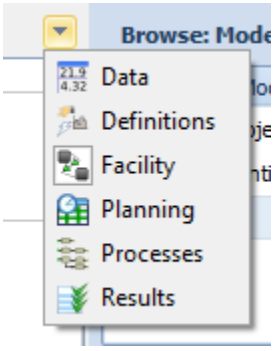


Previously, in this sort of model scenario, when the entity reached Server1 a runtime warning would occur (and the entity's progress then stalled) because the engine currently does not have code capable of figuring out a followable network path to an assigned destination that involves passing through objects (or sub-models).

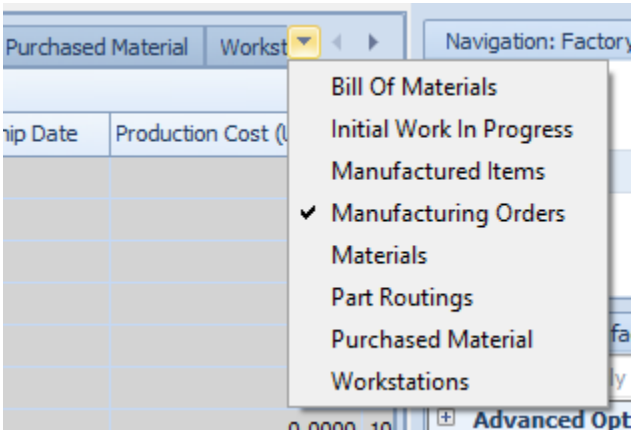
Note that an Advanced Compatibility setting has been added in Run Setup to allow an advanced user to disable this change, as well as guarantee than any models built in Simio Sprint Version 97 or earlier will continue behaving exactly as before.

Sketchup Support
We now support the latest version of Trimble Sketchup, Sketchup 2013, which was released in May 2013. Prior to this sprint of Simio, we supported only versions 3through 8.
User Interface Tab Enhancements
We have changed our tab views and table views to use the DevExpress docking windows enhancements.

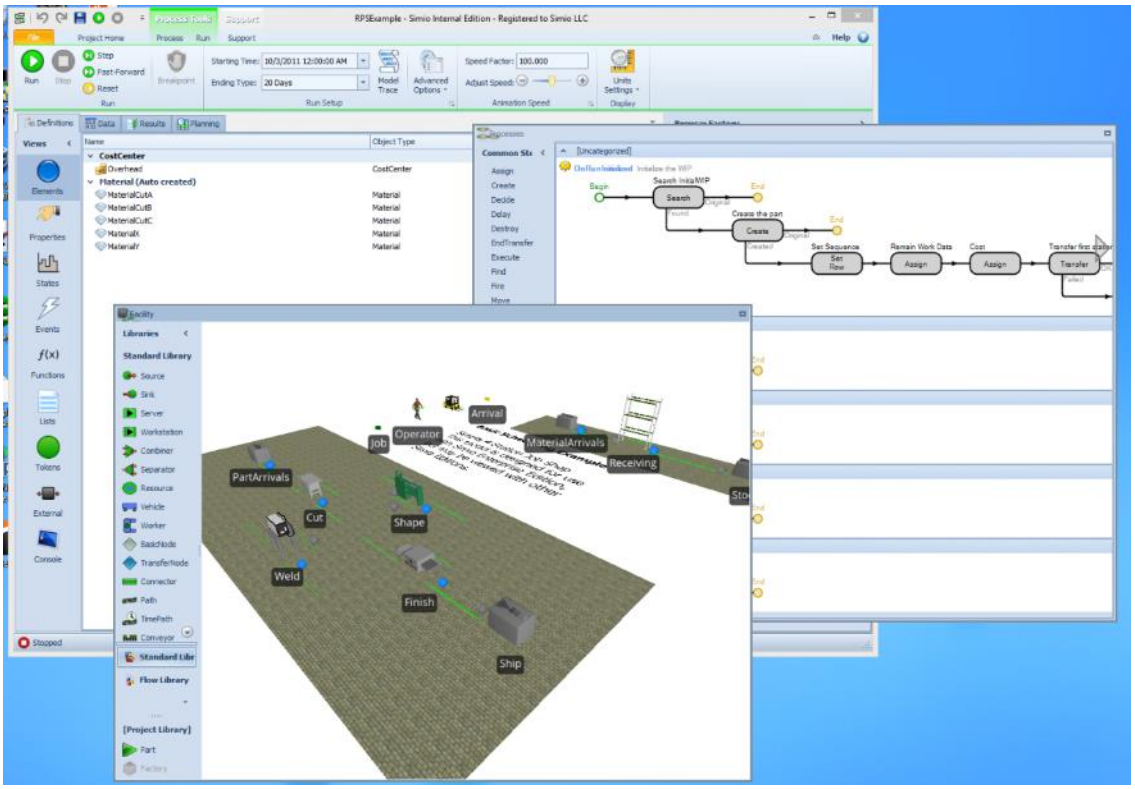
To the far right of the main tabs (Facility, Processes, Definitions, Data, etc.) you will see a small drop down arrow on the side to quick jump between tabs:



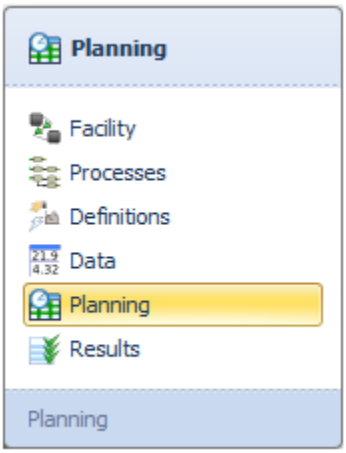
This feature will be useful in our table view with models with lots of tables:



You can also float the tab as a separate top level window by double clicking on it, right clicking and selecting float or simply clicking and moving it to another part of the screen. You can then maximize that window as well if you so choose...



Control-Tabbing with the Tab button gives you a dialog as well that can be used to quickly move between the various windows:



Simio Release 6 – Sprint 97 – November 15, 2013

In this release, we have enhanced our RPSExample model and added two new SimBits demonstrating our new Dashboard Reports features. We've also made some stabilization enhancements, as noted below, and bug fixes. We are excited about all the new features that have made it in to Simio 6, as noted in the last several sprints' release notes sections – we hope you are too!

Dashboard Reports SimBits

We have added two new SimBits that display the new Dashboard Reports features available to Simio Enterprise and Team users.

- **DashboardReportTallies.spfx –**This SimBit takes the original TallyStatisticsInTables project and includes a dashboard report plot for one of the entity type tallies in the system.
- **DashboardReportInteractiveResults.spfx –**This SimBit is a revision of the VehicleVisitsServiceCenter project that graphically displays the resource utilization as well as a vehicle's pickup and drop off times in chart format.

Updated RPS Example

We have updated the RPSExample.spfx project to include:

- Forklift added to transport material between material arrivals, receiving and stock.
- Server symbols updated to be more realistic.
- Worker changed to be an animated person.
- Dashboard reports added including Dispatch List and Order Details.

We have also relaxed the limits in the academic enterprise versions so the RPSExample model will now load and run without deleting any targets.

Table Performance Enhanced

Opening a grid view (or expanding a detail view) that has a foreign key column where there were a lot of foreign key selections should now be much faster. Combo drop downs in general should be faster as well and you should no longer see 'flashing' as you open a combo with a large number of items.

Properties of Top Level Model

The properties of a top level model are streamlined -- many that were rarely used are now hidden. You can still get to them by modifying their visibility in the model's properties window.

Memory Usage for States

We now use much less memory for non-scalar states (e.g., vectors, matrix).

Simio Release 6 (Beta) – Sprint 96 – November 1, 2013

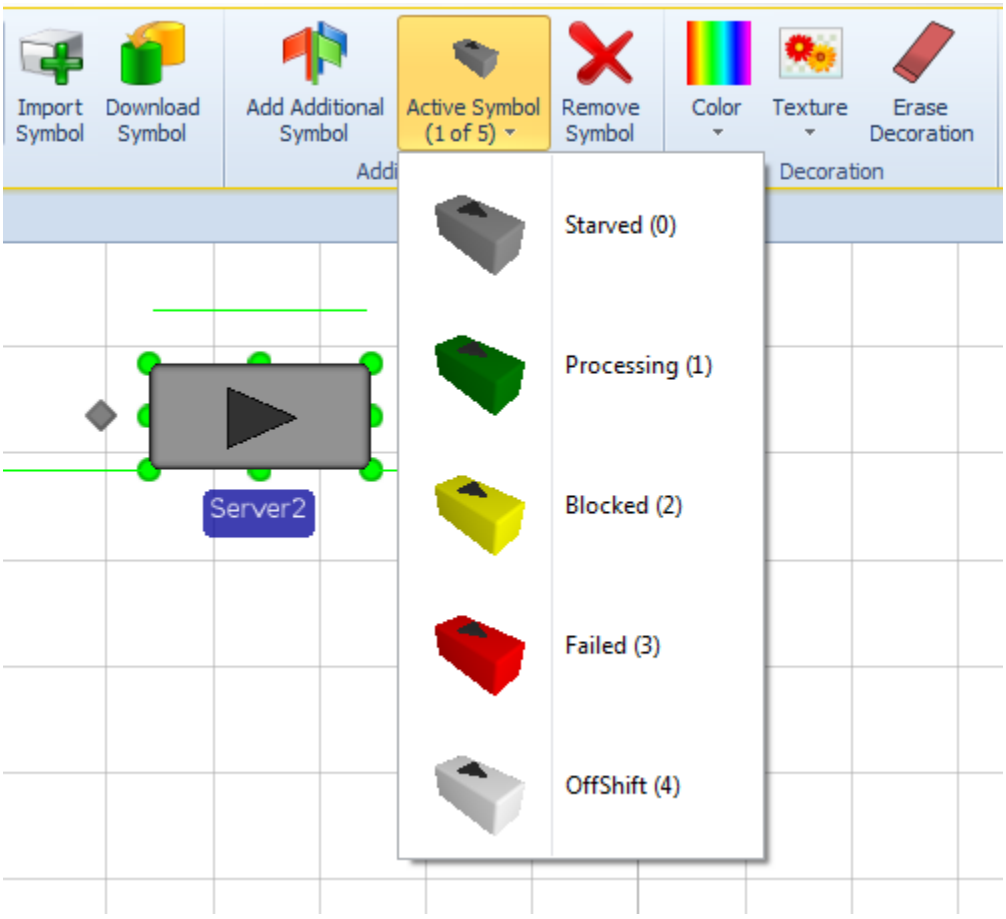
In this release, we added many valuable features. We have enhanced our Flow Library to include an Emptier and added a new Dashboard feature for greater output analysis. We've added a number of new features based on customer requests, including table property functions, multiple symbols for Standard Library objects, and improved entity-viewing features to name a few.

This software is the final Beta release of Simio 6 software and we appreciate any early feedback you may have.

Multiple Symbols on Standard Library and Flow Library Objects

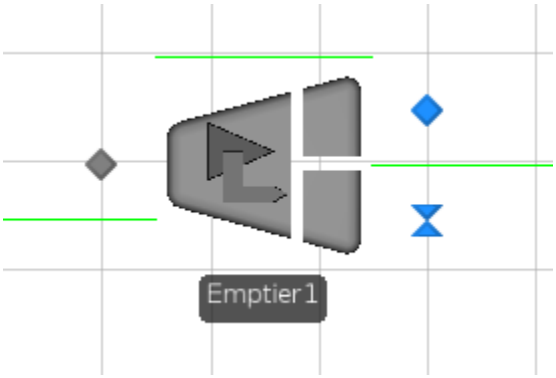
We have added pre-defined multiple animation symbols for many of the objects within the Standard and Flow Libraries. When you place an object, such as a Server, you will now see that there are multiple symbol values associated with that object, each with a distinguishing color.

For example, with a Server, there are 5 states including Starved, Processing, Blocked, Failed and Offshift. The default symbol value (0) is grey, but the Server automatically changes to green when the symbol value is 1 (Processing), 2 (Blocked), 3 (Failed) and 4 (OffShift). This is based on *the Current Symbol Index* for the Server, which is 'Server.ResourceState' by default.



If you resize the first object in the list, Simio will automatically resize all the corresponding objects. If you change the symbol from the default 'block' to something like a drill, for example, Simio will automatically change all associated pictures to the drill symbol, using the same color as the original (we do not change colors of new symbol pictures).

New Flow Library Object – Emptier



The Emptier object may be used to model an emptying process that removes flow product from container entities. The flow rate of the Emptier is regulated by its 'FlowOutput' flow node.

The Emptier is similar to the Server and Filler objects in that it has 3 stations, including the InputBuffer, Processing and OutputBuffer stations. Container entities may be emptied until they are completely empty or until a specific quantity has been removed. As with many objects in the Standard Library, the Emptier object may have a work schedule, failures, state assignments and multiple add-on process triggers.

New SimBits

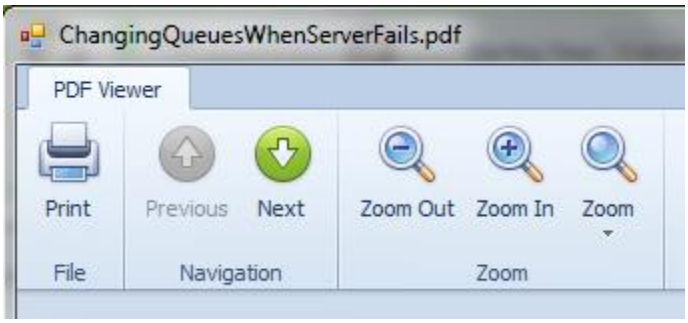
We have added two (2) new SimBits to our extensive SimBit library:

- **DisableFailureTimers.spfx –**This SimBit illustrates how to turn off Failures in a model that runs only until a given number of entities in the system has been processed. Typically, failure events will keep a simulation model running even though there may be no other events occurring in the system.
- **KeepQueueTimeForLast10Entities.spfx –**This SimBit demonstrates the use of storage queues to hold entity information while the actual entity is processing elsewhere. Copies of the most recent 10 entities in a queue are held in a storage queue for calculating a moving average type wait time statistics.

Evaluating Seize Request for Server, Combiner, Separator, Workstation, Filler, and Emptier

Like the Worker and Resource objects, there is now an 'Evaluating Seize Request' add-on process trigger available for Server, Combiner, Separator, Workstation, Filler, and Emptier. These processes can be used when the object is evaluating whether to accept or reject a request to seize capacity of the particular object. In the executed decision process, assigning a value of less than or equal to '0' to the executing token's ReturnValue state (Token.ReturnValue) indicates that the seize request is rejected. This feature allows the server-type object to do "smart" seizes like other resource objects without having to subclass them.

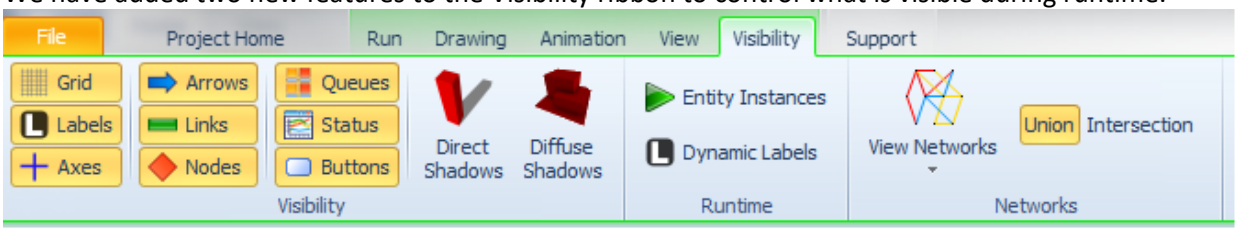
Pdf Viewer for SimBits



We have added a new pdf viewer for displaying pdf file associated with any model files. This interface allows zooming in/out as well as printing. This allowed us to continue to show our pdf files on the desktop, as various windows operating systems were handling pdf viewers differently.

Entity Instances and Dynamic Labels on Visibility Ribbon

We have added two new features to the Visibility ribbon to control what is visible during runtime.

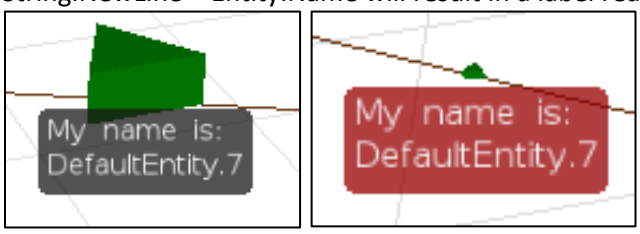


The **Entity Instances** icon will enable users to visibly turn on and off the Entity and Transporter instances placed in the Facility window during the simulation run. For old models, this icon is enabled and will show all instances of Entity and Transporter objects in the Facility window. For new models, this icon is initially disabled.

The **Dynamic Labels** icon works in conjunction with a new property on Entities and Transporters (which includes Worker / Vehicle) named *Dynamic Label Text* (in the Animation category of properties). The *Dynamic Label Text* is an expression which returns a string to show a floating label below a dynamic entity at runtime. The **Dynamic Labels** icon on the Visibility ribbon allows users to enable or disable these labels separately from the (static) Labels. This floating label is similar to the label that can be placed in the background, in that is always “faces” the viewer and is always visible regardless of zoom level unless obstructed by another label or object. It is particularly useful when you have models covering a large scale and you want to locate an entity (for example see truck traffic on miles of roadway).

String.NewLine Function

The String.NewLine function will allow you to now put various strings together and a line is generated between the strings. This new string function is particularly useful in Write steps or with the Dynamic Labels described above. For example, within a Dynamic Label Text step, the expression "My name is:" + String.NewLine + Entity.Name will result in a label readable regardless of zoom level:



Settings Button on File Menu

We have added a Settings option on the main File menu which allows users to specify their graphics card for displaying 3D graphics (currently includes DirectX and Open GL). These application settings also include options to load (or suppress loading) the Standard Library and Flow Library, plus an option to specify multiple additional user libraries to load upon opening Simio. For example if you seldom use the Flow Library, but frequently use the Crane Library, you can reflect that here for automatic loading.

New Regulator Functions

OutputFlowReceivers.NumberItems - Returns the current number of entities receiving outflow from the regulator.
OutputFlowReceivers.FirstItem - Returns a reference to the first entity in the list of entities currently receiving outflow from the regulator.
OutputFlowReceivers.LastItem - Returns a reference to the last entity in the list of entities currently receiving outflow from the regulator.
OutputFlowReceivers.IndexOfItem(entity)- Returns a reference to the entity at a specified index position in the list of entities currently receiving outflow from the regulator.
OutputFlowReceivers.ItemAtIndex(index)- Returns the one-based index of a specified entity in the list of entities currently receiving outflow from the regulator. If the entity is not a current output flow receiver of the regulator then the value 0 is returned.
OutputFlowReceivers.Contains(entity)- Returns True (1) if the list of entities currently receiving outflow from the regulator includes the specified entity. Otherwise, the value False (0) is returned.

Table Property Functions

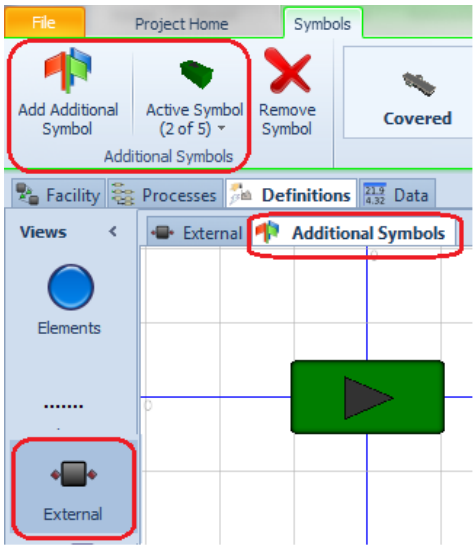
In previous software versions, the object/model could have a Table type property. However, there were some restrictions on when this property could be used. We now support of the following table property syntax:

TableProperty[row, column] - Returns a value from a row, column location of a table pointed to by a table property. NOTE: row and column must both resolve to integer values.
TableProperty.AvailableRowCount - Returns the number of rows for a table pointed to by a table property.
TableProperty.TimeIndexedRow - Returns the time indexed row of a table pointed to by a table property.

Some examples of using these table properties include the following:

TableProperty1[3, 4] returns the value at row 3, column 4 within the table referenced in TableProperty1.
MyFixed1.MyTableProp[3, 4] returns the value of row 3, column 4 within the table referenced in MyTableProp within some instance of an object named MyFixed1

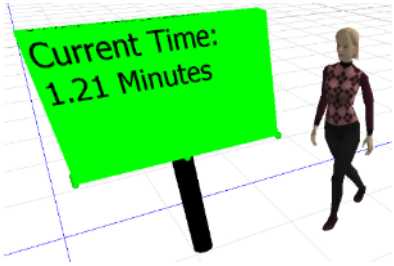
Ability to Assign Additional Symbols for User Defined Objects through External Panel



The multiple symbols on library objects feature previously described is available because of the new capability to assign additional symbols for an object's external view through the Definitions tab / External panel of the object.

Now, instead of just an External window, there is an Additional Symbols tab that will allow users to define the look, color and texture of multiple symbols for that object definition (not just the object instance when it is placed in a model).

Object Height on Status Objects

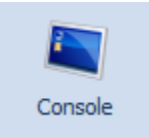


Object Height (on the Drawing ribbon) now applies to Status Label, Floor Label, Status Pie, Circular Gauge and Linear Gauge. If any of these have a non-zero height, they will be drawn as a box, with the same texture on the two largest sides. This is useful for creating a billboard or upright status information display as with the sign on a post illustrated here.

Support for Nautical Distances and Speeds

If you are modeling shipping systems you will appreciate that Simio now supports length units of Nautical Miles and speed units of Nautical Miles per Hour (commonly known as a "knot"). For you land lubbers, a nautical mile represents 1 minute of longitude at the equator and 1 knot is about 1.15 mph.

Dashboard Tab on Main Model/Object Changed to Console in Definitions Tab

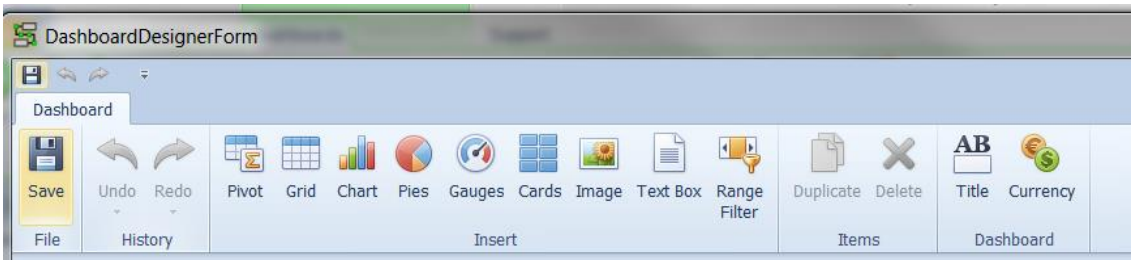


We have replaced the Dashboard tab within the main tab group (Facility, Processes, Definitions, Data, Results) with a Console panel option in the Definitions window. This Console window functions the same as the Dashboard window previously and includes the Animation ribbon for placing status displays in the window. Default Console windows are available for the Source, Server, Sink, Workstation, Combiner and Separator objects and can be accessed by right-clicking on the object and selecting the Open Console option.

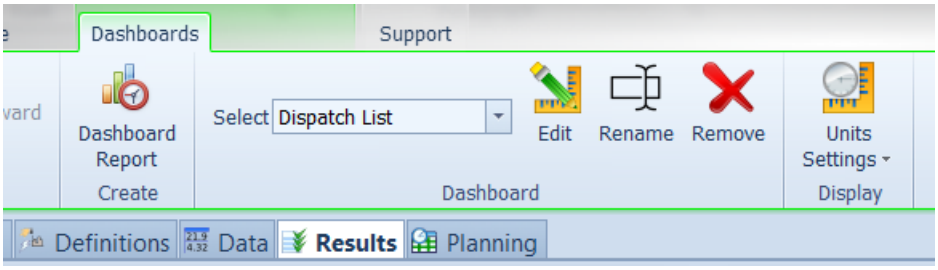
Similar to the previous Dashboard window, items within the Console window of the Definitions tab can be seen during the simulation run by tiling the windows such both the Facility window and Console panel of the Dashboard window area displayed.

Dashboard Reports

Dashboard *reports* (not to be confused with the recently renamed Dashboard *tab*, discussed above) allow you to display a dashboard based report of generated model data. A dashboard allows you to organize and present model information that is customizable and easy to read. Model data can be displayed in many graphical forms such as grids, charts and gauges. It also allows you to customize your dashboard by adding images, text boxes and filter ranges.

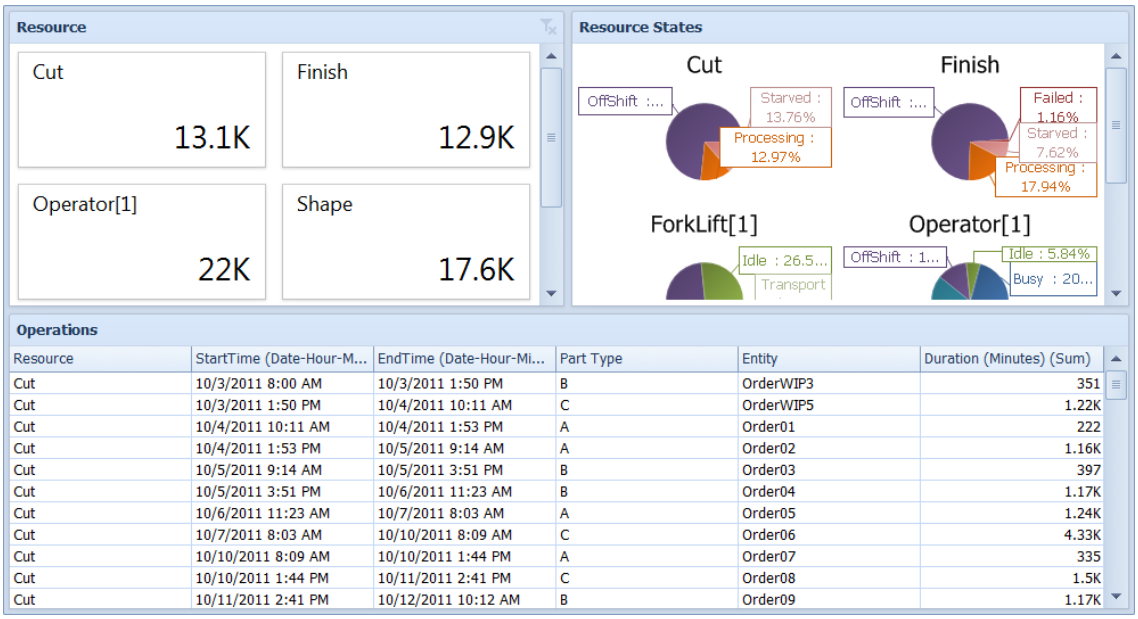


The major components of the Dashboard Report tool include creating a dashboard report and selecting a dashboard report to view, edit, rename or remove. There is also the option to set display settings such as time, length and distance, volume and mass and weight.



Depending on your license type, you can create and/or view dashboards that contain state and tally observations, user defined tables or other interactive logs:

- All editions can view previously created dashboards.
- Team or Academic edition is required to create new dashboards.
- Enterprise edition is required for full dashboard capability.



Operations

Resource	StartTime (Date-Hour-M...	EndTime (Date-Hour-Mi...	Part Type	Entity	Duration (Minutes) (Sum)
Cut	10/3/2011 8:00 AM	10/3/2011 1:50 PM	B	OrderWIP3	351
Cut	10/3/2011 1:50 PM	10/4/2011 10:11 AM	C	OrderWIP5	1.22K
Cut	10/4/2011 10:11 AM	10/4/2011 1:53 PM	A	Order01	222
Cut	10/4/2011 1:53 PM	10/5/2011 9:14 AM	A	Order02	1.16K
Cut	10/5/2011 9:14 AM	10/5/2011 3:51 PM	B	Order03	397
Cut	10/5/2011 3:51 PM	10/6/2011 11:23 AM	B	Order04	1.17K
Cut	10/6/2011 11:23 AM	10/7/2011 8:03 AM	A	Order05	1.24K
Cut	10/7/2011 8:03 AM	10/10/2011 8:09 AM	C	Order06	4.33K
Cut	10/10/2011 8:09 AM	10/10/2011 1:44 PM	A	Order07	335
Cut	10/10/2011 1:44 PM	10/11/2011 2:41 PM	C	Order08	1.5K
Cut	10/11/2011 2:41 PM	10/12/2011 10:12 AM	B	Order09	1.17K

Gantt Visibility Option (Enterprise only)

There's a setting on each entity instance called "Gantt Visibility", an expression that is evaluated for each row in the Resource Usage Log, Transporter Usage Log, and Constraint log, to determine if this row should appear in the Entity Gantt chart.

Transporter Usage Log (Enterprise only)

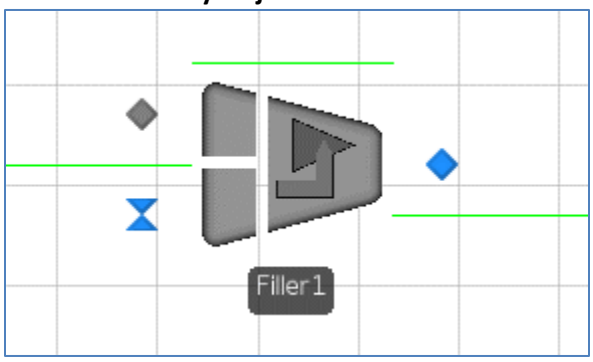
The Transporter Usage Log now has "From Node" and "To Node" columns.

Simio Release 5 – Sprint 95 – October 8, 2013

In this release, we have enhanced our Flow Library to include two additional objects -- a Filler and a ContainerEntity, as discussed below. We've also added a number of SimBits to our extensive SimBit library, including some work flow (WF_*) type examples and several models displaying our new Flow Library objects.
This software continues to be a 'Beta' version of Simio 6 software and we appreciate any early feedback you may have.

New Flow Library Object - ContainerEntity
The ContainerEntity object may be used to model a type of moveable container (e.g., barrels or totes) for carrying flow entities representing quantities of fluids or other mass.
ContainerEntities include a container element that may include initial contents within the container. Similar to a ModelEntity, the ContainerEntity may have travel and routing logic (initial desired speed, network, priority and sequence). Additional features, such as state assignments and add-on processes when the container is full and empty, are also available.

New Flow Library Object - Filler



The Filler object may be used to model a filling process that fills container entities with flow entities representing quantities of fluids or other mass. The filling flow rate of the Filler is regulated by its 'FlowInput' flow node. Container entities enter the Filler through its 'ContainerInput' basic node and exit the Filler from its 'Output' transfer node.

The Filler is similar to the Server in that it has 3 stations, including the InputBuffer, Processing and OutputBuffer stations. Container entities may be filled until they are full or to a specific quantity. As with many objects in the Standard Library, the Filler object may have a work schedule, failures, state assignments and multiple add-on process triggers.

New SimBits

We have added four new SimBits to our SimBit library:

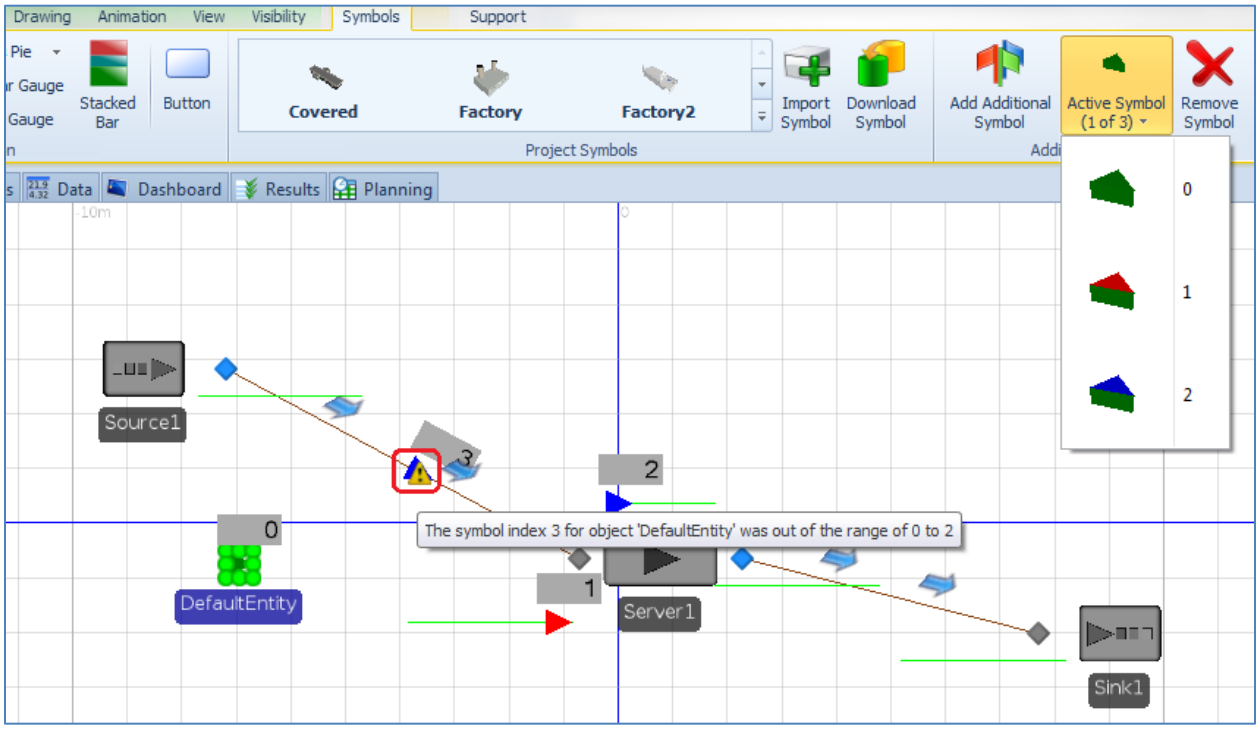
- **ChangingQueuesWhenServerFails.spfx –**This SimBit shows the logic involved for searching an input queue of a server when the server fails to remove and redirect entities to another server for processing.
- **CheckingEntityAheadOnLink.spfx –**This SimBit uses the EntityAheadOnLink function on a link to evaluate and make decisions on entity link spacing.
- **WF_AdditionalResource.spfx –**This SimBit illustrates a work flow environment where multiple entity types require a different number of resources for processing and functions are used for processing times.
- **WF_Authorization.spfx** – This SimBit illustrates the use of multiple sets of workers within a work flow environment such that various types of entities require workers with different sets of skills for processing.

We have also updated the **FlowConcepts** SimBit to include a new model:

- **SimpleFiller** – This model illustrates the use of the new Filler and ContainerEntity objects within the Flow Library. ContainerEntity objects enter the Filler as discrete entities, while flow from a FlowSource enters the Filler object as flow type entities. The containers are filled and a full ContainerEntity exits the Filler to a Sink.

Revised Warning for Animation Symbols – Out of Range

In prior Simio versions, if an object's Current Symbol Index value was 'out of range' based on the value assigned and symbols defined, a large warning would appear and temporarily halt the simulation model. We have now changed this behaviour such that if an object's animation symbol index is not within the defined range, an animated exclamation point, as shown below, is shown over the symbol with a warning displayed when the animation is paused and mouse hovered over the object. After several user's feedback on the original method, we have changed this warning to be less obtrusive and hopefully more suitable to all users.



Simio Release 5 – Sprint 94 – September 18, 2013

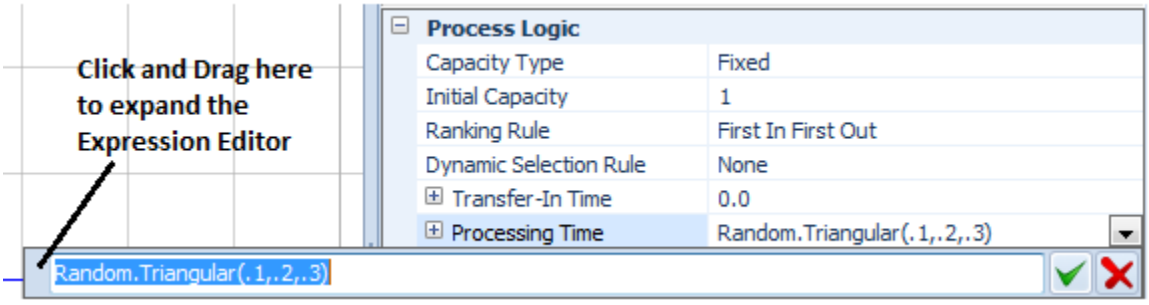
In this sprint release, we have added many miscellaneous features that have been requested by our users over time. We are also continuing to work toward Simio Release 6, which will be released later this fall. You may wish to consider this a 'Beta' version of that software, with much more to come. We appreciate any early feedback you may have.

Flow Library Enhancement

The FlowNode object (and associated nodes within the FlowSource and Tank objects) has been enhanced to include a *Split Allocation Rule* of 'Evenly If Possible' for splitting flow on the output flow from the node. When this rule is used, the flow regulator will distribute flow as evenly as possible to the node's outbound links while maximizing total outflow.

Expression Editor Resize

The Expression Editor can now be resized to allow for viewing/input of larger expressions. To resize the editor, simply click and drag the left portion of the expression editor and resize it. This size will remain for all expression editing until it is changed by the user.



Experiment Window Use of Control – Category Name

When model properties are defined, they may be specified into different categories, using the *Category Name* field on the property. We now utilize these category names when displaying the properties within an experiment as well.

Profile Reporting for Run Performance Evaluation

Within the Advanced Options button of the Run ribbon, there is a new option to Enable Run-Time Profiling. If this option is enabled, during interactive runs Simio will record execution counts and timing for various parts of the model. The results are summarized in the Profile window, which can be opened by selecting the Profile button in the windows area of the Project Home ribbon. This will allow users to evaluate where performance improvements to their model may be made.

Profile

Item	% Time in Parent	% Time Overall	Count	Plot
[Calendar Event]	99.98	99.9771	80702	
Path.OnReachedEnd.FromPath [Transfer]	28.11	28.1014	9230	
[Simulation Engine]	19.75	19.7436	80702	
Worker.OnVisitingNode.Worker [UnPark]	4.40	4.3983	1383	
Server.OnEnteredOutputBuffer.ToOutputNode [Transfer]	4.04	4.0365	1473	
OnRunInitialized.InitializationTable [Search]	2.81	2.8069	1	
Bed.OnEnteredProcessing.SecondaryResource [Release]	2.37	2.3713	381	
TransferNode.OnEntered.ToOutboundLink [Transfer]	2.11	2.1115	1206	
Bed.OnEnteredProcessing.SecondaryResources [Seize]	1.91	1.9062	381	
Server.OnEnteredProcessing.ToOutputNode [Transfer]	1.76	1.7613	755	
Worker.OnVisitingNode.Worker [Release]	1.66	1.6601	501	

On Interval Process Option for Time-Indexed Tables

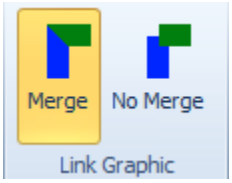
Within a Time-Indexed table (Advanced Options properties on the table), the optional *On Interval Process* is now available. When the index into the table is updated based on the time-interval specified, the On Interval Process is run at the start of each interval for any variable updating or calculations that a user may wish to make.

Gesture Support in 3D for Touch Screen Users

Pinch-to-zoom and two-finger-rotate gestures are now available within the 3D view, provided the user has a touch screen.

Merge/No Merge Options on Appearance Ribbon for Connecting Links

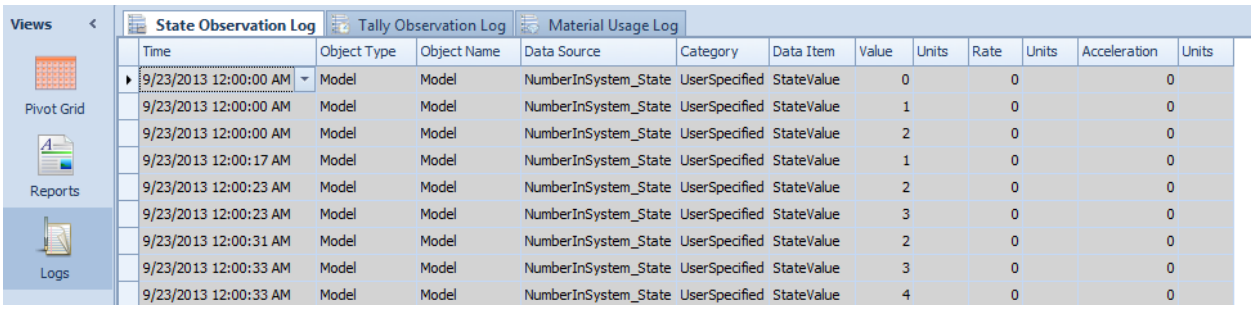
When a Node is highlighted, the Appearance ribbon now includes Merge and No Merge options for graphically showing the links that intersect at the node. Merge will smoothly connect the two paths, while No Merge will show a slight disconnect. This is primarily used when the paths coming into and out of the node include path decorators.



New Logs within the Results Tab

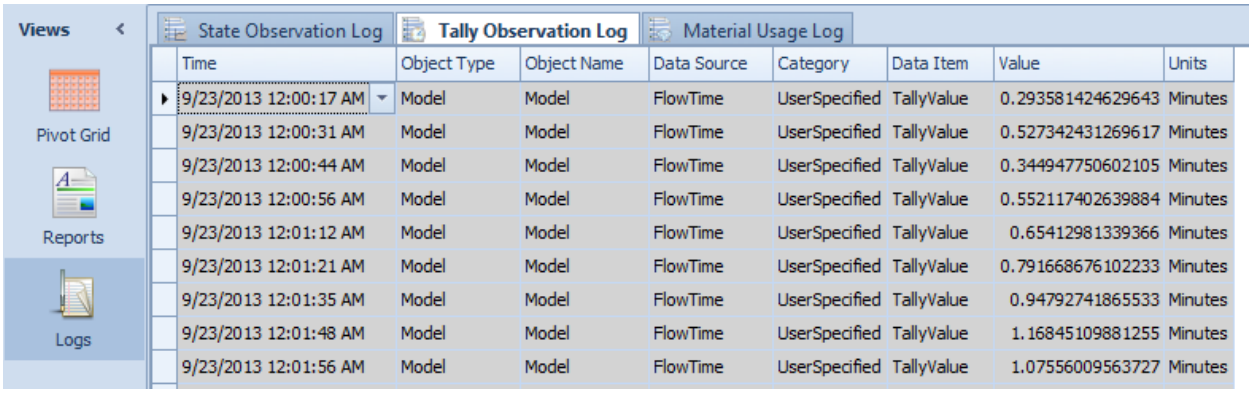
We now have three new logs that provide detailed information regarding state variables, tally statistics and material usage. For each of the State Statistic, Tally Statistic and Material elements, there is a new *Log Observations / Log Material Usage* option that can be specified as 'True' to enable logging.

The State Observation Log provides state variable information for those State Statistic elements that have logging specified as 'True'. When the value of a state variable or expression changes, that value is recorded in the log.



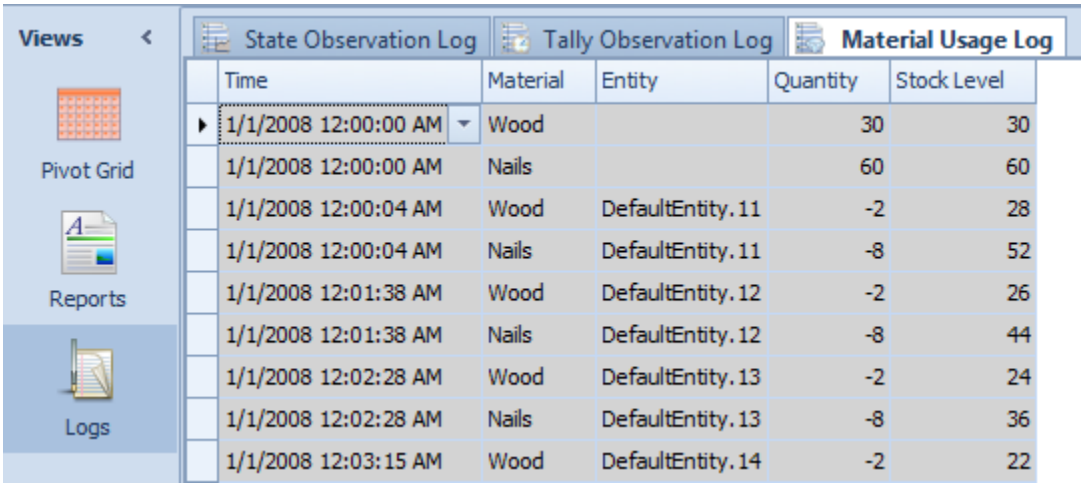
Time	Object Type	Object Name	Data Source	Category	Data Item	Value	Units	Rate	Units	Acceleration	Units
9/23/2013 12:00:00 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	0		0		0	
9/23/2013 12:00:00 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	1		0		0	
9/23/2013 12:00:00 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	2		0		0	
9/23/2013 12:00:17 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	1		0		0	
9/23/2013 12:00:23 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	2		0		0	
9/23/2013 12:00:23 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	3		0		0	
9/23/2013 12:00:31 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	2		0		0	
9/23/2013 12:00:33 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	3		0		0	
9/23/2013 12:00:33 AM	Model	Model	NumberInSystem_State	UserSpecified	StateValue	4		0		0	

The Tally Observation Log displays Tally Statistic observation values during the simulation run.



Time	Object Type	Object Name	Data Source	Category	Data Item	Value	Units
9/23/2013 12:00:17 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.293581424629643	Minutes
9/23/2013 12:00:31 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.527342431269617	Minutes
9/23/2013 12:00:44 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.344947750602105	Minutes
9/23/2013 12:00:56 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.552117402639884	Minutes
9/23/2013 12:01:12 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.65412981339366	Minutes
9/23/2013 12:01:21 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.791668676102233	Minutes
9/23/2013 12:01:35 AM	Model	Model	FlowTime	UserSpecified	TallyValue	0.94792741865533	Minutes
9/23/2013 12:01:48 AM	Model	Model	FlowTime	UserSpecified	TallyValue	1.16845109881255	Minutes
9/23/2013 12:01:56 AM	Model	Model	FlowTime	UserSpecified	TallyValue	1.07556009563727	Minutes

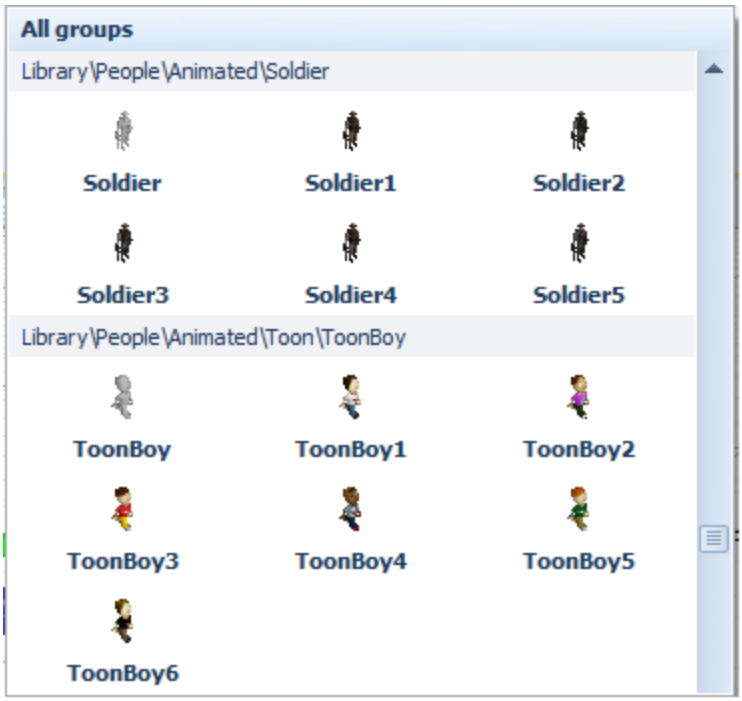
The Material Usage Log displays information on materials that are produced and consumed during the simulation run.



Time	Material	Entity	Quantity	Stock Level
1/1/2008 12:00:00 AM	Wood		30	30
1/1/2008 12:00:00 AM	Nails		60	60
1/1/2008 12:00:04 AM	Wood	DefaultEntity.11	-2	28
1/1/2008 12:00:04 AM	Nails	DefaultEntity.11	-8	52
1/1/2008 12:01:38 AM	Wood	DefaultEntity.12	-2	26
1/1/2008 12:01:38 AM	Nails	DefaultEntity.12	-8	44
1/1/2008 12:02:28 AM	Wood	DefaultEntity.13	-2	24
1/1/2008 12:02:28 AM	Nails	DefaultEntity.13	-8	36
1/1/2008 12:03:15 AM	Wood	DefaultEntity.14	-2	22

New Animations for Moving Objects (Walking People)

We have added a number of new libraries of 'animated' moving symbols. Prior to this sprint, we had animated males and females. Our animations now additionally include children, elderly persons, soldiers and toons (for fun ☺).


All groups
Library\People\Animated\Soldier
Soldier
Soldier1
Soldier2
Soldier3
Soldier4
Soldier5
Library\People\Animated\Toon\ToonBoy
ToonBoy
ToonBoy1
ToonBoy2
ToonBoy3
ToonBoy4
ToonBoy5
ToonBoy6

Simio Release 5 – Sprint 91 – July 16, 2013

In this sprint release, we have added several new features designed to help our beginning users, in addition to some new functions and SimBits for more advanced users. Within the Properties window of an object, you will now see a Show Commonly Used Properties checkbox (initially enabled) that will expose only those commonly used properties within an object. Of course, all of our features and flexibility are still there and can be accessed by unchecking this checkbox.

Show Commonly Used Properties – Option for New Users

At the top of the Properties window of an object, you will now see the checkbox *Show Commonly Used Properties*.

When this mode is enabled (the default), the display is limited to the key set of properties that defines the core behavior of each object. Many standard features such as failures, state assignments, secondary resource allocations, financials, and custom add-on process logic that are provided by the Standard Library objects are not displayed. This mode allows the beginner to focus on the key concepts for each object without any additional complexities.

When this option is disabled (unchecked), the full set of properties for each object is exposed. Note that this mode does not directly affect model behavior or results. It is just hiding/un-hiding properties but not changing property values. This mode is typically used when first learning Simio and then disabled once the basic concepts are mastered.

Elements Functions for Intelligent Objects

Every object now has an 'Elements' namespace that provides generic access to the collection of direct child elements contained within that object. These new Elements type functions may be used, for example, in a Find step to search all the direct child elements of an object to find a particular element and/or condition.

Elements.NumberItems- Returns the number of direct child elements contained within this parent object.
Elements.FirstItem - Returns a reference to the first element in the list of direct child elements contained within this parent object.
Elements.LastItem - Returns a reference to the last element in the list of direct child elements contained within this parent object.
Elements.ItemAtIndex(index) - Returns a reference to the element at a specified index position in the list of direct child elements contained within this parent object.
Elements.IndexOfItem(element) - Returns the one-based index of a specified element in the list of direct child elements contained within this parent object. If the element is not a direct child element of this parent then the value 0 is returned.
Elements.Contains(element) - Returns True (1) if the specified element is a direct child element of this parent object. Otherwise, the value False (0) is returned.

Enhanced Documentation

The e-book formerly named Introduction to Simio has new content including an introduction to simulation, simulation success skills, and an enhanced glossary. It has been renamed and is available on the Support ribbon.

New Videos

Two new video sets (Flexible Manufacturing Series and Learning Simio Lab Series) have been adding to the existing Introduction to Simio course and all three are now directly available from the Support ribbon.

New SimBits

We have added five new SimBits to our SimBit library:

- **ChooseAlternateSequence.spfx –**This SimBit shows entities that have a typical main sequence of server steps, but due to inspection / rework process, some go through an alternate sequence and then return to the main sequence.
- **MergingConveyorsControlledByGate.spfx –**This SimBit uses a Resource object as a type of 'gate' to control the sequence of entities arriving from two merging conveyors.
- **VisitAllServersInAnyOrder.spfx –**This SimBit illustrates how a simple modification can be made to an object (Server) to permit more intelligent use, allowing selection of only objects that still need the service it provides. This is done by sub-classing a Server and adding a state variable for tracking the entities that have processed.
- **ExcelReadWrite.spfx** – This SimBit illustrates how to read and write using an Excel spreadsheet file.
- **DBReadWrite**.spfx – This SimBit illustrates how to read and write using a MySQL database.

Enhanced Button Animation

The Button on the Animation ribbon now has a *Text Scale* property to resize the text size on the button. This was requested by multiple users recently.

Model Properties

Model Properties has continues to be available via a left click menu, but you will see that it will now be displayed at other times, like when you click on the background of the facility view. The contents have now been better organized as well to prominently display only the properties of most interest. If you have not previously discovered it, when you create reference properties, they appear under a special "Controls" category in model properties. This makes a convenient interface for working interactively with a model.

Expression Editor

You will now find that the list of items on the Expression Editor pull down list is much shorter. We have now suppressed display of many items that are rarely used to make the frequently used ones more prominent.

Simio Release 5 – Sprint 89/90 – June 24, 2013

In this sprint release, we have added some subtle advanced features for object building and fixed a number of software issues that have been recently discovered. The frequency and content of our public releases has temporarily slowed while we concentrate our development resources on a major new feature. Watch for news on this coming soon.

Container Element 'Empty' and 'Full' Events
We have added the Empty and Full events to the Container element. These events are fired when a container is empty or full, respectively. This allows other objects to take some event-triggered action when a container completes a fill or empty operation.

Monitor Element Enhancement
We have enhanced the State Variable Name property of the Monitor element to allow the specific state variable that is to be monitored to be a member of an object or element reference state variable thus allowing the state variable monitored by an individual Monitor element to be dynamically changed. A Monitor element will dynamically 'rewire' to the state variable to be monitored each time the Monitor is enabled (i.e., the Enabled state changes from False to True).

For example, you can now monitor a state variable like MyEntityReferenceState.SomeState where MyEntityReferenceState is a state variable assigned to point to some entity. This is valid as long as the Monitor is always Disabled whenever the MyEntityReferenceState variable is set to Nothing. So, typical usage would be to disable Monitor, assign reference state to point at whatever entity you want the Monitor to monitor a state value on, and then enable the Monitor.

Custom Object Enhancement
You can now create a custom object that has an 'Input' external node that is inputting entities into a station, node, or container location that is specified as a state variable reference. So, you can create a new custom fixed object, go to the External panel view and add an External Node symbol, then specify the input location as a state variable. Then, dynamically assign that state variable during the run (depending on whatever logic) to have entities go into whatever location the state variable is currently assigned.

Consistency in NaN evaluations
NaN stands for "Not a Number" and is the value taken on by a numeric field when it does not have a valid numeric value. Examples include the average value of a statistic before any observations have been recorded or an expression like "Math.Sqrt(-1)". We have improved the consistency of Largest/Smallest value selection rules and Minimize/Maximize expression searches when evaluating candidates with NaN values. We now ignore or discard those candidates that result in NaN (not a number) based on the expression specified.

Table Binding Enhancement

Shift-Click on the Remove Binding button within the Data window tables will ask the user if they would like to remove binding for all tables. This allows you to 'unbind' all tables with binding at one time. This is particularly useful when you are sharing a model with someone else and you wish to run the model using only the built-in data tables.

Simio Release 5 – Sprint 88 – May 14, 2013

In this sprint release, we've added a number of new user defined steps and elements to provide the capability to read and write to databases and Excel spreadsheets. In addition, we have added a few right-click options in different areas to quickly add breakpoints in the Processes window and to quickly add Links between recently placed nodes. Some of our other new features are for user flexibility, including allowing more options on the transfer related steps/objects and providing local culture arguments for string and datetime functions.

'Specific' and 'Outbound Link Name' Options on Transfer Step, BasicNode and TransferNode

We have added the 'Specific' option to the Transfer step, BasicNode and TransferNode for indicating the outbound link preference when transferring an entity out of a node. The Outbound Link Name property then allows the user to specify the link, either by entering the link directly or, more typically, by referencing a link reference state.

Examples of using this new feature may include using a global 'Switch' state variable that is being dynamically assigned to 'switch' outbound flow direction from the node according to some designed control system logic.

Alternatively, perhaps each entity is executing decision logic (using steps such as Find, Search, or Decide) to do dynamically determine which outbound link it wants to select, store that selected link in a state variable, and then transfer out from the node using that selected link by configuring the 'Specific' outbound link preference on the node.

Yet another example is perhaps a property or state variable on an entity that is specifying which outbound link to use. This property might be based on entity type, or perhaps a variable that has been dynamically assigned based on where the entity is in its processing sequence or if it is 'rework' etc.

String and DateTime Function Enhancements

For any of the below functions, if the useLocalCultureInfo argument is 'False' or unspecified, then the comparison will be culture insensitive.

String.Compare(string1, string2 [, useLocalCultureInfo]) - The useLocalCultureInfo argument indicates whether to use local culture-specific information to influence the string comparison.

String.CompareIgnoreCase(string1, string2 [, useLocalCultureInfo]) - The useLocalCultureInfo argument indicates whether to use local culture-specific information to influence the string comparison.

String.FromDateTime (dateTime [, formatString, useLocalCultureInfo]) - The datetime value may be converted to its equivalent string representation using a specified date and time format string. The formatString argument should contain either a single format specifier character (see .NET Standard Date and Time Format Strings) or a custom format pattern (see .NET Custom Date and Time Format Strings). The useLocalCultureInfo argument indicates whether to convert the datetime value to a string using local culture-specific format information.

String.ToDateTime (dateTime [, useLocalCultureInfo]) - The useLocalCultureInfo argument indicates whether to convert the string to a datetime value using local culture-specific format information.

String.FromReal (value [, formatString, useLocalCultureInfo]) - The numeric value may be converted to its equivalent string representation using a specified numeric format string. The formatString argument should contain either a valid standard numeric format specifier (see .NET Standard Numeric Format Strings) or any combination of custom numeric format specifiers (see .NET Custom Numeric Format Strings). The useLocalCultureInfo argument indicates whether to convert the numeric value to a string using local culture-specific format information.

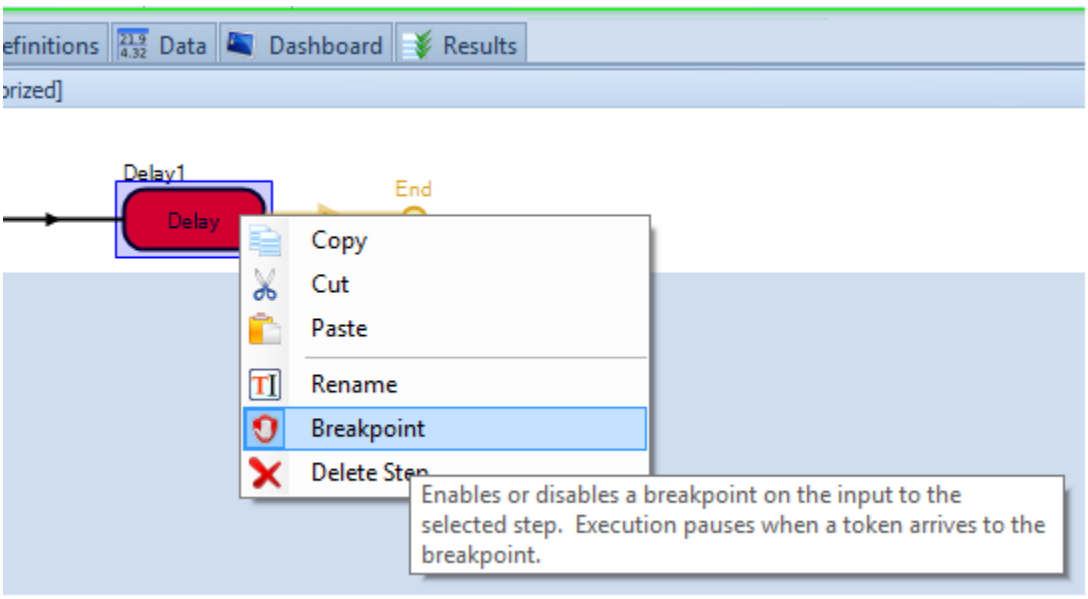
String.ToReal (string [, useLocalCultureInfo]) - The useLocalCultureInfo argument indicates whether to convert the string to a numeric value using local culture-specific format information.

DateTime.ToString (dateTime [, formatString, useLocalCultureInfo]) - The datetime value may be converted to its equivalent string representation using a specified date and time format string. The formatString argument should contain either a single format specifier character (see .NET Standard Date and Time Format Strings) or a custom format pattern (see .NET Custom Date and Time Format Strings). The useLocalCultureInfo argument indicates whether to convert the datetime value to a string using local culture-specific format information.

DateTime.FromString (string [, useLocalCultureInfo]) - The useLocalCultureInfo argument indicates whether to convert the string to a datetime value using local culture-specific format information.

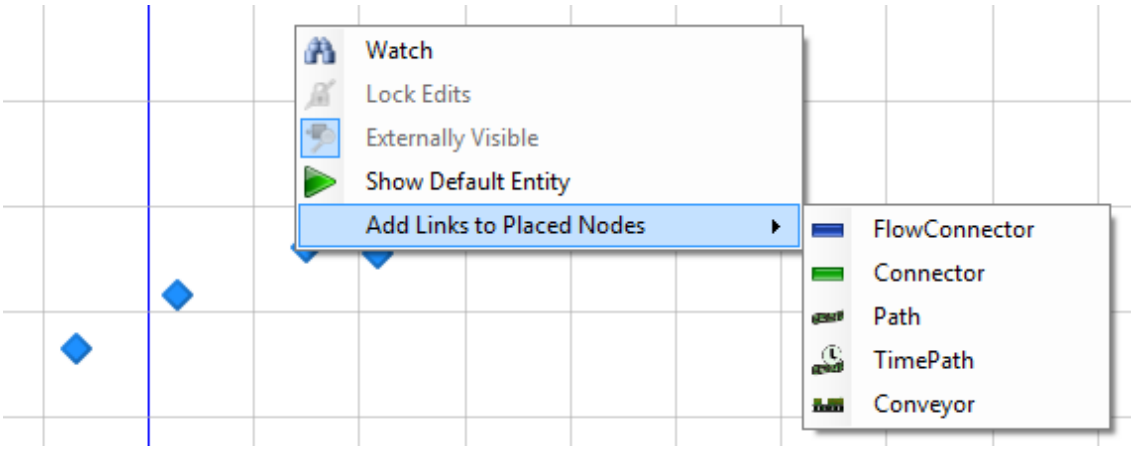
For Steps in Processes Window – New Breakpoint Option on Right-Click

Now, in addition to using the Breakpoint icon on either the Process or Run ribbons to set a breakpoint on a particular step, you can highlight the step and use the right-click menu, Breakpoint option, to turn on and off a breakpoint.



Ability to Add Links to Multiple Nodes Quickly

We have added a new feature to be able to quickly add links to a group of nodes that have been placed in the Facility window. Now, if the last two or more undo-able actions you did were placing nodes, you will see the 'Add Links to Placed Nodes' menu item in the right-click menu within the Facility window. This will allow you to create links of a specified type between the nodes that were just placed. For example, if you double click on the Standard Library object TransferNode to place multiple nodes and then click and place 5 nodes in the Facility window, you can right-click, select the new menu item (Add Links to Placed Nodes), select the link type (Path) and Simio will connect each of the first 4 nodes with a Path to the node that was subsequently placed.



After the links get created, they are all selected as well, so you can do bulk edits to them in the property grid.

Note that this only shows up if the last undoable action was placing a node. So if you place 3 nodes, then go to, say, the table view and create a table, then come back to the Facility window, you *will not* see this option, as the last undoable action you did was create a table.

Common Steps Panel Reorganized

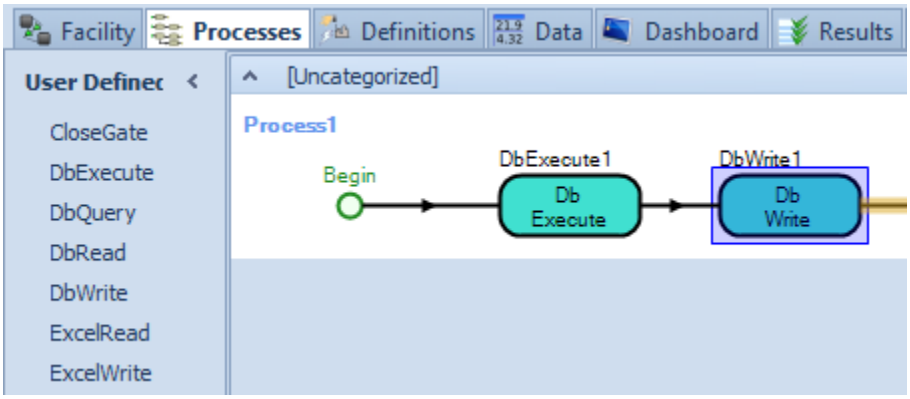
The steps within the Common Steps panel are now listed in alphabetical order instead of 'logical' type order. This was done based on several customer requests, as well as internal discussions on ease of use.

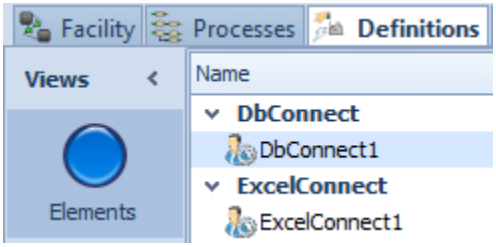
New User-Defined Steps and Elements for Database and Excel

We have added the six new user defined steps and two corresponding user defined elements for reading and writing to databases and Excel spreadsheets.

Within the User Defined panel in the Processes window, the database related steps include DbExecute, DbQuery, DbRead and DbWrite. The DbExecute step is designed to execute an SQL statement on a database. The DbQuery step may be used to query data from a database. The DbRead and DbWrite steps allow user to read from and write to a database. These steps require a corresponding element, the DbConnect user defined element, which specifies the database connection string and provider information.

Also within the User Defined panel of steps, the ExcelRead and ExcelWrite steps have been added to read values from and write values to an Excel spreadsheet. The corresponding element is the ExcelConnect, which specifies the Excel workbook information required.





Simio Release 5 – Sprint 87 – April 20, 2013

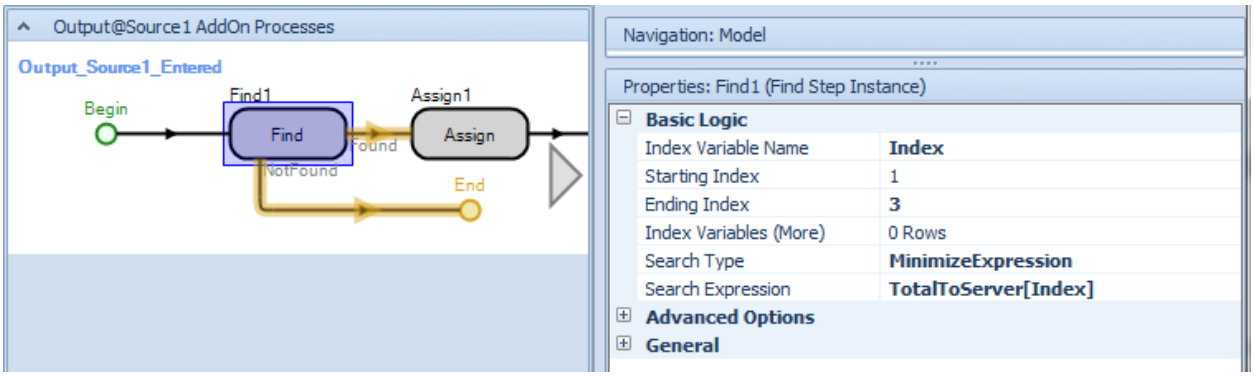
In this sprint release, we have added many user requested enhancements, including a much requested step to be able to search a state variable or array, the **Find step**. This allows users to evaluate values within a state array or matrices for a minimum or maximum value or one that meets a certain condition. It also allows for finding values within a table (similar to the Search step), except the Find step only returns the first found value.

Other user requested features include a new SimBit for flow, transparent option on floor labels, more flexibility in table and schedule referencing, and additional options on symbol import and database binding.

New Find Step

We have added a new Find step to the Processes window that may be used to search the value of an expression over a specified range of one or more indexed variables. The expression will typically involve array state variables (vectors or multi-dimensional arrays) or indexing related functions.

The Find step may be used to search a vector type state variable for a minimum or maximum value, for example, or to search the array for a given condition (values over the amount of X). This is a feature that many customers have asked for, as the Search step does not search for values within state variables.



New SimBits

We have added two new SimBits to our extensive SimBit library:

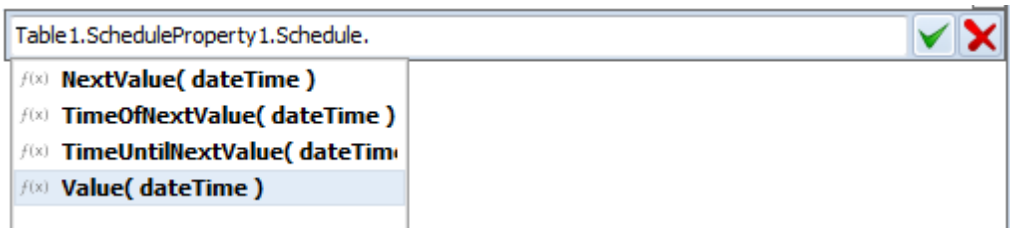
- **FindAMinimumStateValue.spfx –** This SimBit illustrates the use of the new Find step to search a state variable array for the minimum value.
- **PickUpDropOffFlow.spfx** – This SimBit demonstrates the Flow Library features in conjunction with an entity that has a container. The entity's container is filled at one location and then the entity travels to a destination where it empties its container.

Table Referencing Enhancement

We have added the ability to reference a data table entry as **TableName[row, column]** in an expression. This makes it easier, for example, to use Find step to search a table in the same manner as a variable matrix.

Schedule Referencing Enhancement

A Schedule property now provides access to the specified schedule data. This schedule property may be a property that is passing some schedule data into an embedded object's logic or may be a property within a table. Within the expression editor, you will now see for any schedule property access to the schedule functions.



Note that if you attempt to reference the schedule data stored in a schedule property, but no schedule name value has been specified for the property, then an 'Object or element reference not set to an instance of an object' error will be thrown by Simio. To avoid this error, the 'HasValue' function provided by any property (also a Sprint 87 new feature, seen below) may be used to do a conditional check that a property has a value before attempting to access the object or element reference stored in it.

Transparent Option on Floor Label

We have added the option for a transparent type background to floor labels. This was a user requested enhancement.

Symbol Rotation

We have added the option for rotating a symbol 180 degrees in the Symbol Import window. This was a user requested enhancement.

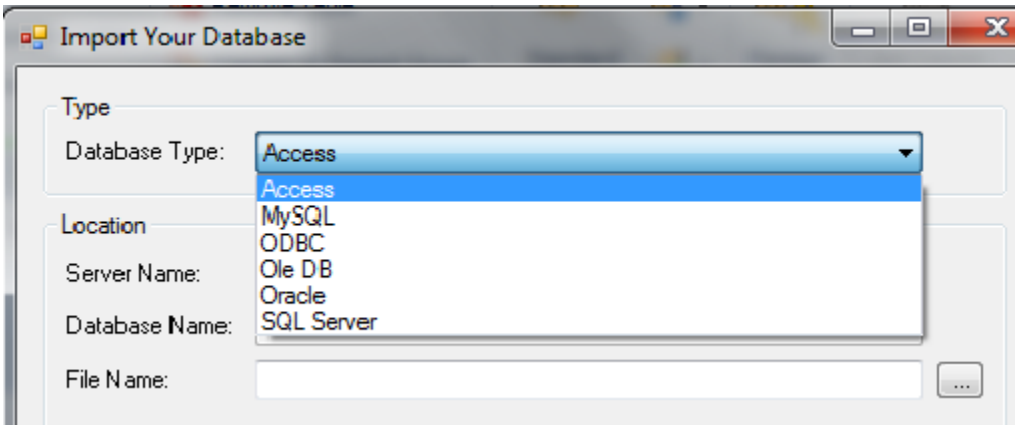
HasValue Function for Element and Object Properties

A new **HasValue** function has been added for all Element and Object type properties in the model. The function can be referenced as 'ElementPropertyName.HasValue' or 'ObjectPropertyName.HasValue'. This is useful for checking model logic whether an optional property has been specified a value or not. The function HasValue returns True(1) if a value has been specified for the property. Otherwise the value False(0) is returned.

** Repeat Group Note – If the property is a repeat group, HasValue will return True(1) if one or more entries have been entered into the repeat group. Otherwise, the value False(0) is returned.

Database Binding Options – MySQL

Currently, there are many database type options when binding your table data to a database. We have now added the MySQL option to this list:



Performance Improvements

We have made some changes internally for run speed improvement that will impact some but not all models.

Simio Release 5 – Sprint 86 – April 4, 2013

In this sprint release, we have continued to enhance the Processes window steps to allow for specifying a particular object when performing a given step. These enhancements allow you to write process logic without requiring that the executing token is associated with the object that the step is acting upon, or requiring the process steps to have to be inside the 'Parent' object that they are acting upon.

We are sure you will enjoy our latest animation feature – walking people. In the software, we have provided a number of male and female animated symbols, located in Library\People\Animated in the Project Symbols area. These animated symbols have associated animations that allow them to walk, run, jump, swim and many other actions. Users can also import their own animations, if desired.

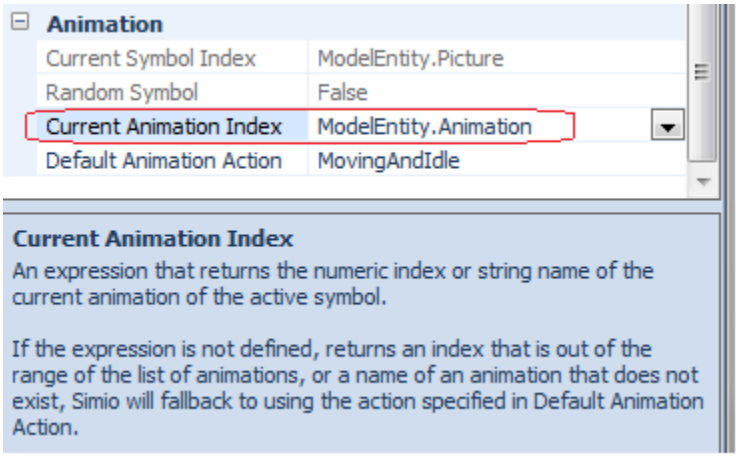
'Entity Type' and 'Entity Object' Enhancement on Batch, UnBatch, PlanVisit, SelectVisit, and Arrive Steps

The PlanVisit, SelectVisit, Batch, UnBatch and Arrive steps now support a 'SpecificObject' choice to specify the specific entity object to batch, unbatch, search visit requests, select destination of a visit request or notify visit request of arrival. All steps now include an *Entity Object* property for specific entity referencing and are no longer limited to just ParentObject or AssociatedObject choices for the *Object Type* property.

Animated Objects (Walking People)

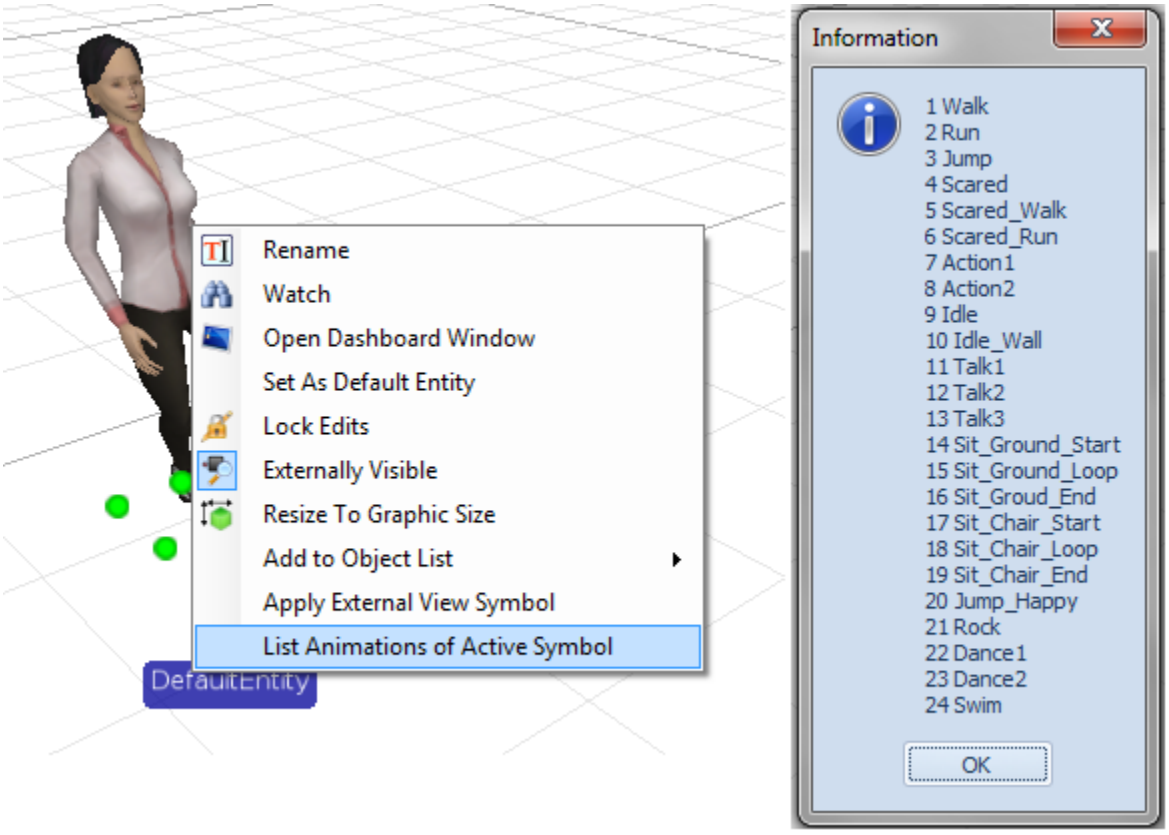
We have enhanced our animation support to include animations, which provide more realistic movement of entities through the system. Simio includes multiple male and female symbols within the Library\People\Animated area of the project symbols.

The ModelEntity within the Project window now has a string state named Animation, referred to within the model as ModelEntity.Animation. This string state is then used within the ModelEntity properties window as the default value for the Current Animation Index (see below).



Once an entity is placed in the Facility window, the default green arrow symbol can be replaced by an animated person (from the Library\People\Animated directory within the Project Symbols). Right-

clicking on an animated object will allow you to select *List Animations of Active Symbol*, which in turn will open the list of animations for that particular object, as shown below.



The animations may be referenced by either numeric index (in the case above, 1 through 24) or by string value (i.e., “Walk”). When using the ModelEntity.Animation string state, the string values above should be used. For example, in an Assign step upon arrival within a Source, the ModelEntity.Animation may be set directly to “Run” or could be assigned to be Random.Discrete(“Run”, .25, “Dance1”, .5, and “Swim”, 1). In that case, 25% of the entities would run, 25% would dance and the other 50% would swim.

Alternatively, the Current Animation Index property itself may reference either the string or number directly (instead of using the default value of ModelEntity.Animation).

The entity’s speed, ModelEntity.DesiredSpeed, will also impact the entity movement (i.e., a slower moving entity will appear to be walking, while a faster moving entity be running).

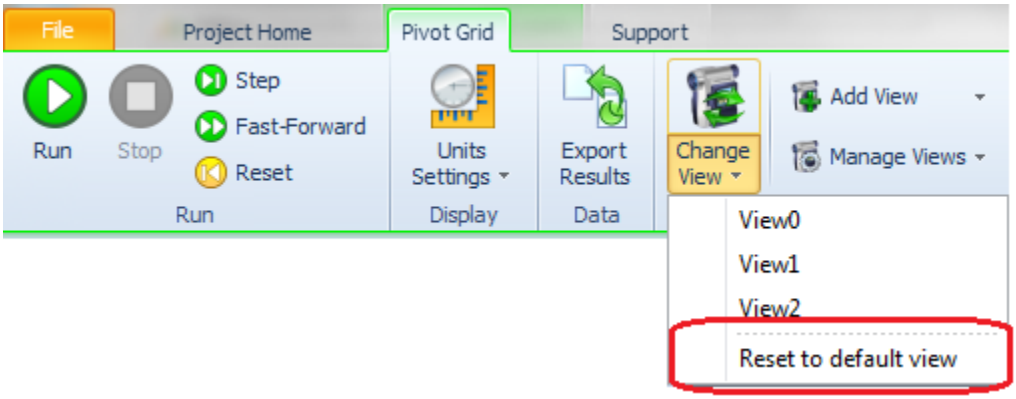
New Animation SimBit

We have added a new SimBit to our SimBit library:

- **AnimatedPeople.spfx –** This SimBit includes a demonstration of various ways to animate moving people (entities) in the system.

New ‘Reset to default view’ in Pivot Grid Window

In addition to being able to add, manage and change views within the Pivot Grid window, you may now also reset the pivot grid table to the original default view.



Simio Release 5 – Sprint 84/85 – March 14, 2013

In this sprint release, we have added many new functions and enhanced multiple steps to make the product more flexible. See the note below for examples of using these new features.

Additionally, several items have been added to the Simio Enterprise Edition to provide additional information to the end user on transporters and schedules, including new logs and Gantt changes.

Finally, we've started work on animated people and this sprint will provide a glimpse at that capability. Documentation, help and examples of this functionality will be provided within the next sprint, but check out the ThemePark example for a quick look!

General Note about New Step Enhancements and Functions

Enhancements made to many of the steps (as noted below) are simply to allow you to write process logic without having to be required that the executing token is associated with the object that the step is acting upon, or requiring the process steps to have to be inside the 'Parent' object that they are acting upon. This flexibility may be particularly useful to object builders. Many of these features, for example, are used to simplify modeling within the Crane library.

'Controller' process logic external to a vehicle, for example, can now use Pickup/Dropoff steps to command the vehicle to pick up or drop off entities. Or, you could model a train where all the pickup/dropoff logic is inside a single 'Locomotive' object, and the locomotive's logic is going to direct multiple 'Railcar' transporters pulled by the locomotive to pick up or drop off cargo. Previously, you were forced to put the Pickup and Dropoff steps inside each individual railcar.

Many of the new functions have been added for flexibility as well. You can now use the various schedules functions to determine when a particular resource will be changing shifts and make routing decisions accordingly, for example.

New Schedules Functions

Value(dateTime)- Returns the schedule's value for the specified numeric datetime (simulation time).
NextValue(dateTime) - Returns the schedule's next value if at the specified numeric datetime (simulation time).
TimeOfNextValue(dateTime) - Returns the simulation time (in hours) of the schedule's next value change if at the specified numeric datetime (simulation time).
TimeUntilNextValue(dateTime) - Returns the time duration remaining (in hours) until the schedule's next value change if at the specified numeric datetime (simulation time).

New Resource Object Functions

Capacity.Next - Returns the next scheduled capacity of this resource. *Note: If the resource is not following a work schedule then this function simply returns the current capacity since any future capacity changes are unknown.

Capacity.TimeOfLastChange - Returns the simulation time (in hours) that the capacity of this resource most recently changed.
Capacity.TimeOfNextChange - Returns the simulation time (in hours) of the next scheduled capacity change of this resource.*Note: If the resource is not following a work schedule then this function simply returns infinity since the timing of any future capacity changes is unknown.
Capacity.TimeSinceLastChange - Returns the elapsed time duration (in hours) since the most recent capacity of this resource.
Capacity.TimeUntilNextChange - Returns the time duration remaining (in hours) until the next scheduled capacity change of this resource. *Note: If the resource is not following a work schedule then this function simply returns infinity since the timing of any future capacity changes is unknown.
ResourceOwners.Contains(owner) – Returns True (1) if the list of owners that have currently seized capacity units of this resource includes the specified owner. Otherwise the value False (0) is returned.

New Token Function

AssociatedObject - Returns the object that this token is associated with.

New Entity Functions

NetworkDistanceTo.NextEntityAheadOnLink- If the entity object's leading edge is on a link, then this function returns the distance from the entity's leading edge to the trailing edge of the next entity ahead on the same link. If there is no entity ahead then the value Infinity is returned.
NetworkDistanceTo.NextEntityBehindOnLink - If the entity object's trailing edge is on a link, then this function returns the distance from the entity's trailing edge to the leading edge of the next entity behind on the same link. If there is no entity behind then the value Infinity is returned.
Population.Contains(entity) – Returns True (1) if the population contains the specified entity. Otherwise the value False (0) is returned.
Sequence.DestinationNodes.Contains(node) – Returns True (1) if the entity's assigned sequence contains the specified destination node. Otherwise the value False (0) is returned.

New Queue State Function

Contains(object) – Returns True (1) if the queue contains the specified object. Otherwise the value False (0) is returned.

New Node Functions

InboundLinks.Contains(link) – Returns True (1) if the collection of drawn inbound links into this node contains the specified link. Otherwise the value False (0) is returned.
OutboundLinks.Contains(link) – Returns True (1) if the collection of drawn outbound links from this node contains the specified link. Otherwise the value False (0) is returned.

New Network Element Function

Link.Contains(link) – Returns True (1) if the network's collection of link members contains the specified link. Otherwise the value False (0) is returned.

New Intelligent Object Function

SeizedResources.Contains(resource) – Returns True (1) if the list of resources currently seized and owned by this object includes the specified resource. Otherwise the value False (0) is returned.

'Resource Type' Enhancement on Allocate Step

The Allocate steps now has an additional 'SpecificObject' option for the 'Resource Type' property that that allows the owner that the step's action is in behalf of to be specified as a specific object reference. Before, the owner always had to be either the parent object containing the step or the associated object of the token executing the step. Similar changes were made to many other steps in the previous sprint to provide more flexibility in process logic design.

'Entity Object' Enhancement on Destroy, Route, SetNetwork, SetNode, Park and Unpark Steps

The Destroy, Route, SetNetwork, SetNode, Park and Unpark steps now support a 'SpecificObject' choice to specify the specific entity object to destroy, route, set the network, set the node, park or unpark. All steps now include an *Entity Object* property for specific entity referencing and are no longer limited to just ParentObject or AssociatedObject choices for the *Object Type* property.

'Transporter Type' and 'Transporter Object' Enhancement on Pickup, Dropoff and SelectDropoff Steps

The Pickup, Dropoff and SelectDropoff steps have been enhanced to allow the transporter to be more flexibly specified. Previously, these steps could only be used in process logic of the parent transporter to which the step actions pertained. The *Transporter Type* property allows the user to select the object associated with the executing token, the parent object or a specific object reference. The *Transporter Object* property is visible when the *Transporter Type* is 'SpecificObject' and is the specific object to perform the pickup, dropoff or select dropoff.

'Object' Enhancement on Insert and Remove Steps

The Insert and Remove steps now support a 'SpecificObject' choice to specify the specific entity object to insert into or remove from a queue. These steps now include an *Object* property for specific object referencing and are no longer limited to just ParentObject or AssociatedObject choices for the *Object Type* property.

Transfer Step Enhancement

When transferring from the current node to an outbound link, there are now an *Outbound Link Preference* and *Outbound Link Rule* properties which can override the default settings for those property values that are defined on the node.

Model Warning Added

We now give a warning when the total number of tokens in the system is more than 100,000. We also provide some course detail as to where the problem (if there is one) might be in the model. This will help prevent memory problems when a user may have an undetected logic issue.

Enhance State Arrays/Matrices Dimension Types

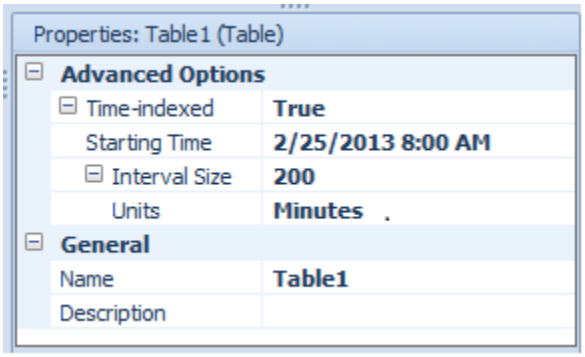
Within states, such as real, integer, object references, element references (all states but Lists), the *Dimension Type* property has been enhanced to include3-dimensional up to 10-dimensional array options. This capability was previously within Simio, however, not as clearly stated.

Time-Indexed Table Properties

A data table may now be specified as Time-indexed, based on the properties of the table, as shown below. When the Time-indexed property is 'True', you may specify the Starting Time and Interval Size. Within a time-indexed table, you can then have a column(s) of values within the table and retrieve a single value from the column, not by direct indexing, but indirectly, based on the current simulation time. Two additional table functions have been added, including:

TableName.PropertyName.TimeIndexedValue – returns the value of the property in the row that corresponds to the current simulation time.

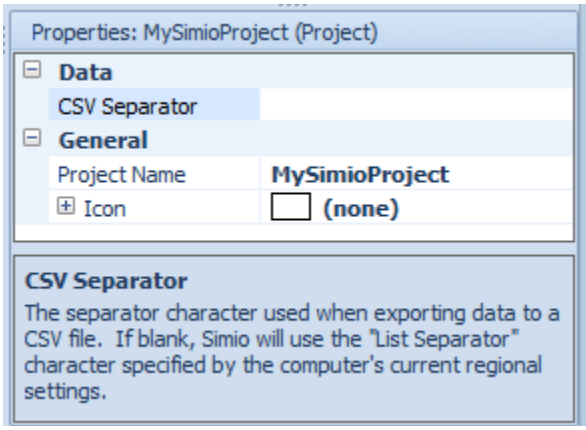
TableName.TimeIndexedRow – returns the row index that corresponds to the current simulation time.



For example, let's say you had a column of values (IntegerColumn1) within Table1 that were 10, 20, 30, 40 and 50, using the above *Interval Size*. Then, from time 0 – 200 minutes, the TableName1.IntegerColumn1.TimeIndexedValue would return a value of 10. For the next 200 simulation minutes, that same function would return the value of 20 and so on.

New CSV Separator for Projects

Within any given project, in Project Properties, there is now a field to indicate the separator character to use when writing to a .csv file.



You can enter a specific character, but the preferred thing to do is to leave it at its default value of blank, and Simio will use the "List Separator" character defined in Windows' regional settings.

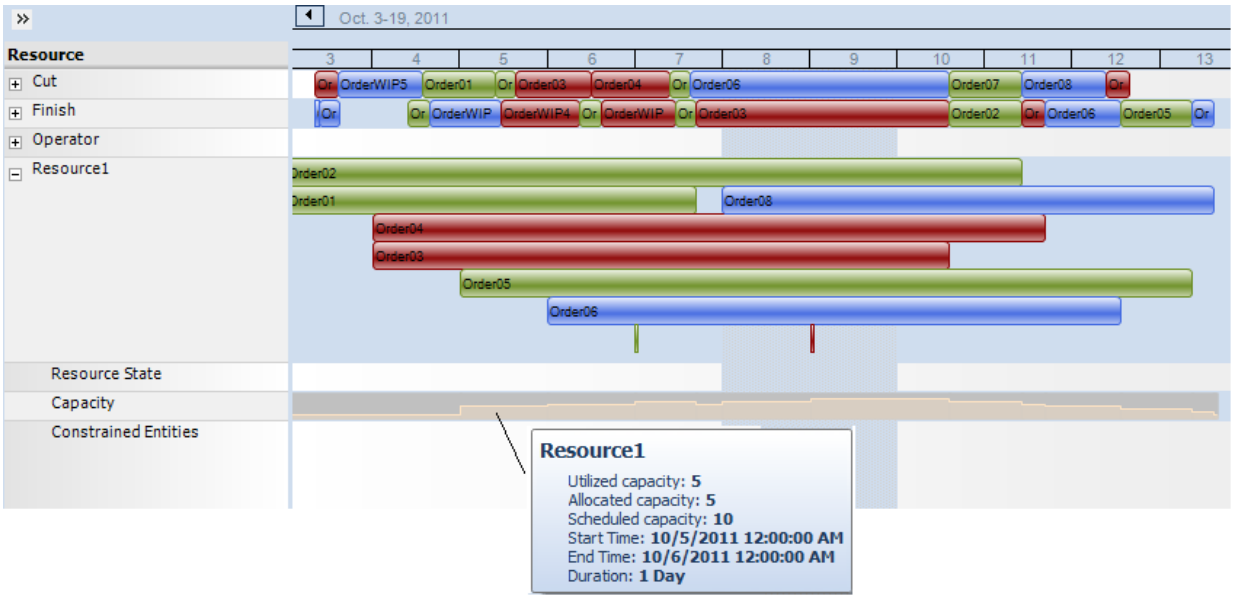
Simio Enterprise Edition – New Resource Capacity Log and Capacity row on Multiple Capacity Resources in Resource Plan Gantt

Within the Simio Enterprise Edition, we have added a Resource Capacity Log to the Logs section of reports. This log displays all of the capacity changes for the resources during the simulation run. This is especially useful when resource capacity is based on a work schedule. The capacity of the resource is displayed, along with the number that has been 'Allocated' (seized), as well as the number 'Utilized'. The utilized value may differ from the allocated value when a failure occurs or when a resource is waiting for material or a secondary resource before processing.

Resource Usage Log | Resource State Log | **Resource Capacity Log** | Constraint Log | Transporter Usage Log

Resource	Scheduled	Allocated	Utilized	Start Time	End Time	Duration (Minutes)
Shape	0	0	0	10/3/2011 12:00:00 AM	10/3/2011 8:00:00 AM	480
Weld	0	0	0	10/3/2011 12:00:00 AM	10/3/2011 8:00:00 AM	480
Cut	0	0	0	10/3/2011 12:00:00 AM	10/3/2011 8:00:00 AM	480
Finish	0	0	0	10/3/2011 12:00:00 AM	10/3/2011 8:00:00 AM	480
Shape	1	0	0	10/3/2011 8:00:00 AM	10/3/2011 12:00:00 PM	240
Weld	1	1	0	10/3/2011 8:00:00 AM	10/3/2011 8:00:02 AM	0.0406667
Operator[1]	1	1	1	10/3/2011 8:00:00 AM	10/19/2011 9:38:29 AM	23138.5
Cut	1	1	1	10/3/2011 8:00:00 AM	10/3/2011 12:00:00 PM	240
Finish	1	1	1	10/3/2011 8:00:00 AM	10/3/2011 12:00:00 PM	240
Weld	1	1	1	10/3/2011 8:00:02 AM	10/3/2011 12:00:00 PM	239.959

Additionally, within the Resource Plan Gantt, when a resource has a capacity greater than one, a new row named Capacity will be displayed and a usage type gantt shown for the capacity allocated for that resource over time. In the below example, Resource1 has a capacity of 10 (gray area). The tan area shows the number of resources allocated over time.



Simio Enterprise Edition – New Transporter Usage Log

Also within Simio Enterprise, we have added a Transporter Usage Log to the Logs reports. This log displays the usage of any transporters (used for transporting, not being Seized). The start time is recorded once the transporter has arrived and loaded the entity and ends once the entity is done being unloaded at its destination. This log corresponds to the transporter's information within the Resource Plan Gantt chart as well.

Resource Usage Log | Resource State Log | Resource Capacity Log | Constraint Log | **Transporter Usage Log**

Transporter	Rider	Start Time	End Time	Duration (Minutes)
Vehicle1[1]	OrderWIP6	10/3/2011 9:17:00 AM	10/3/2011 9:22:09 AM	5.1665
Vehicle1[1]	OrderWIP7	10/3/2011 3:05:09 PM	10/3/2011 3:10:19 PM	5.1665
Vehicle1[1]	OrderWIP5	10/5/2011 9:31:37 AM	10/5/2011 9:36:47 AM	5.1665
Vehicle1[1]	OrderWIP1	10/5/2011 3:19:37 PM	10/5/2011 3:24:47 PM	5.1665
Vehicle1[1]	OrderWIP4	10/6/2011 11:41:13 AM	10/6/2011 11:46:23 AM	5.1665
Vehicle1[1]	OrderWIP2	10/7/2011 8:29:13 AM	10/7/2011 8:34:23 AM	5.1665
Vehicle1[1]	OrderWIP3	10/7/2011 2:50:49 PM	10/7/2011 2:55:59 PM	5.1665
Vehicle1[1]	Order06	10/11/2011 10:10:01 AM	10/11/2011 10:15:11 AM	5.1665

Simio Enterprise Edition – New Gantt Chart Visible/Editable Properties and Gantt Visibility Options

Several smaller items have been enhanced within Simio Enterprise edition including the addition of Gantt Chart *Visible* and *Editable* properties within a data table. This will allow a particular column(s) to be displayed on the Resource Plan Gantt when the table is a Time-indexed type table.

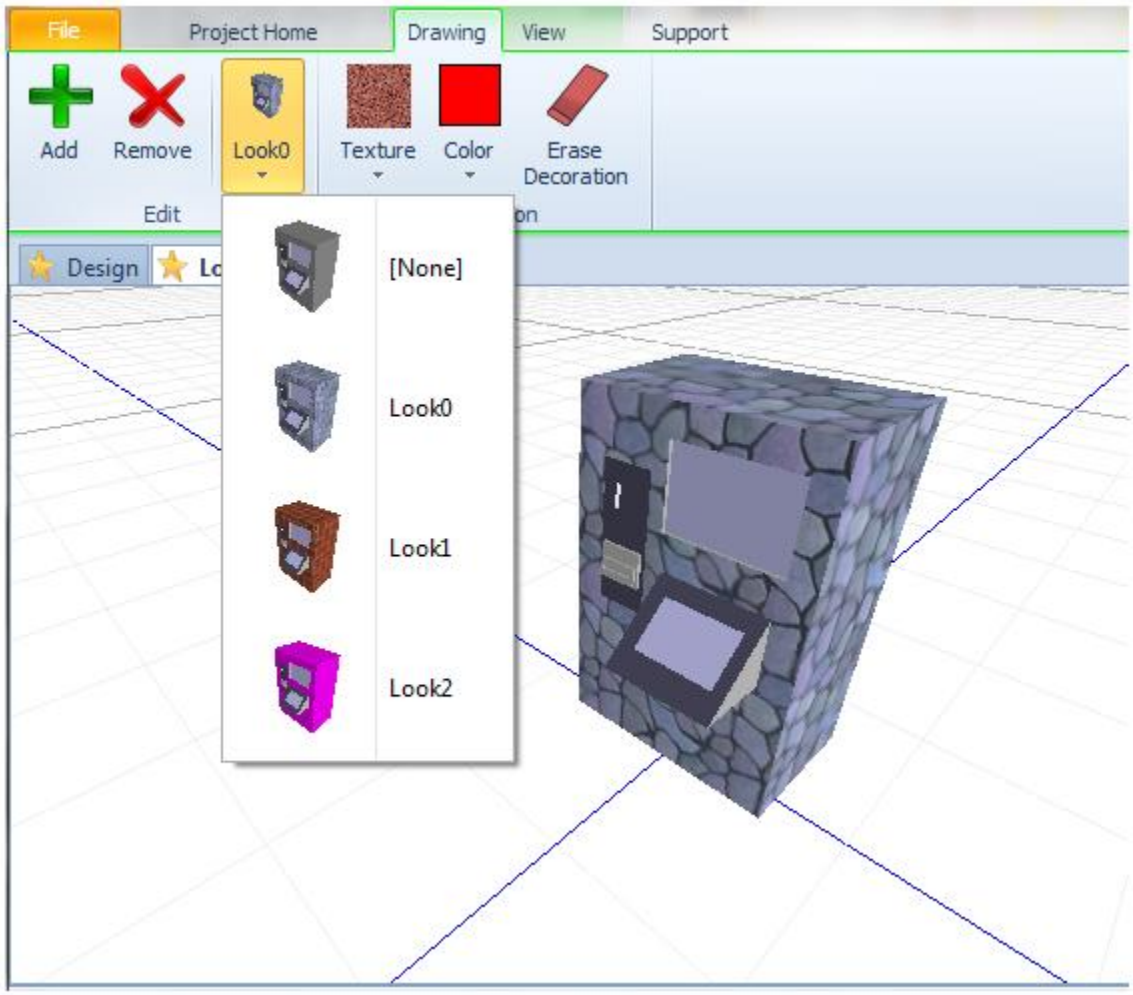
Additionally, various rows may now be toggled on/off within the Resource Plan Gantt by selecting the Gantt ribbon. Rows, such as Resource States, Exceptions and Constraints, can be turned on/off for more specific focus within a particular Gantt chart.

Animated Walking People

Simio now includes walking men and women within the symbol library. When selecting a symbol for an entity, for example, you can now select from multiple types of animated people, found within project symbols in the Library\Animated\People area. Full documentation/help/examples will be introduced in our next sprint. In the meantime, check out the ThemePark example for a demonstration of our latest animation feature!

Additional Symbol 'Look'

Within the Symbol window, we've added a new 'Look' (the previous symbol definition window is now the 'Design' of the Symbol window). For those situations where a single symbol 3d geometry may have several different textures or colors applied to it, the 'Look' allows users to define each of those. These new looks are then put into the library of symbols to be selected for any given project. This saves on both disk and video memory compared to having X different symbols with the same geometry.



Simio Release 5 – Sprint 82/83 – January 31, 2013

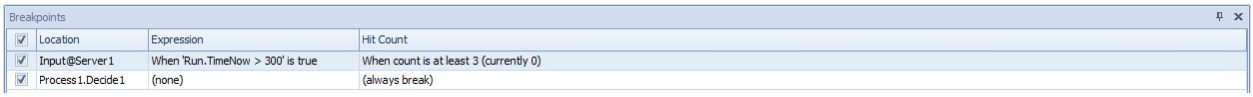
This is our first update after the release of Simio sprint 5 and we think users will be excited about these new flexibility enhancements. We've added a breakpoints window which will allow much greater flexibility in simulation debugging using breakpoints. Multiple functions for nodes and networks, along with increased capability within the Search step will allow users more control over network and link-based logic decisions in their models. Another major enhancement is the addition to many of our steps to allow entity reference states to be used when seizing, releasing, transferring, moving, producing, consuming and searching. And for those users with Simio Enterprise edition or those using the Flow Library, there are several enhancements as well.

New Breakpoints Icon in Project Home Ribbon and Breakpoints Window

A new Breakpoints icon has been added to Windows section of the Project Home window. This opens a Breakpoint window that displays the various breakpoints that have been set by the user in either the Facility or Processes window.

This window also provides additional functionality for turning on / off breakpoints with a checkbox, as well as setting an expression and/or count that must be met for the breakpoint to execute. Expressions may involve simulation time, specific object information (number in a queue), functions, etc. Hit count can be specified as Always, Equal To (count) or At Least (count). The count is not evaluated unless the expression (if any) evaluates to true.

For example, in the screenshot below, the breakpoint on Input@Server1 will not be executed until after 300 time units into the simulation. Then, once the expression is true, the hit count is evaluated. The breakpoint will activate only when the location has been passed at least 3 times (3 or more). A current count is also displayed.



New Entity Function

Population.Name – Returns the string name of the entity's population instance. This is useful when working with just an object instance (e.g., ForkLift5) rather than the object definition (e.g., ForkLift).

New Node Functions

InboundLinks.NumberItems - Returns the number of drawn inbound links into this node.
InboundLinks.FirstItem - Returns a reference to the first link in the collection of drawn inbound links into this node.
InboundLinks.LastItem - Returns a reference to the last link in the collection of drawn inbound links into this node.
InboundLinks.ItemAtIndex(index) - Returns a reference to the link at a specified index position in the collection of drawn inbound links into this node.
InboundLinks.IndexOfItem(link) - Returns the one-based index of a specified link in the collection of drawn inbound links into this node. If the link is not an inbound link into this node then the value 0 is returned.
OutboundLinks.NumberItems - Returns the number of drawn outbound links into this node.
OutboundLinks.FirstItem - Returns a reference to the first link in the collection of drawn outbound links into this node..
OutboundLinks.LastItem - Returns a reference to the last link in the collection of drawn outbound links into this node.
OutboundLinks.ItemAtIndex(index) - Returns a reference to the link at a specified index position in the collection of drawn outbound links into this node.
OutboundLinks.IndexOfItem(link)- Returns the one-based index of a specified link in the collection of drawn outbound links into this node. If the link is not an outbound link into this node then the value 0 is returned.

New Network Element Functions

Links.NumberItems – Returns the number of links that are members of this network.
Links.FirstItem – Returns a reference to the first link in the collection of links that are members of this network.
Links.LastItem – Returns a reference to the last link in the collection of links that are members of this network.
Links.ItemAtIndex(index) – Returns a reference to the link at a specified index position in the collection of links that are members of this network.
Links.IndexOfItem(link) – Returns the one-based index of a specified link in the collection of links that are members of this network. If the link is not a member of this network then the value 0 is returned.

New Functions for any Resource object:

ResourceOwners – The objects that currently own (have seized) capacity units of this resource.
ResourceOwners.NumberItems – Returns the number of objects that currently own (have seized) capacity units of this resource.
ResourceOwners.FirstItem – Returns a reference to the first owner in the list of owners that have currently seized capacity units of this resource.
ResourceOwnters.LastItem – Returns a reference to last owner in the list of owners that have currently seized capacity units of this resource.
ResourceOwners.ItemAtIndex(index) – Returns a reference to the owner at a specified index position in the list of owners that have currently seized capacity units of this resource.

ResourceOwners.IndexOfItem(owner) – Returns the one-based index of the first occurrence of a specified owner in the list of owners that have currently seized capacity units of this resource. If the owner has not seized the resource then the value 0 is returned.

Search Step Enhancements

Search step enhancements have been added to allow the searching of new collection types including TransporterList, NodeInboundLinks, NodeOutboundLinks, NetworkLinks, ResourceOwners, and ObjectPopulation. Link-related searching may be useful for special node-oriented process logic as part of logical decision making.

The Search step also now provides both Starting Index and Ending Index properties to narrow the range of items to search within the collection. Previously, only a Starting Index option was provided. The Starting Index and Ending Index properties are also now available for the 'MinimizeReturnValue' and 'MaximizeReturnValue' objective-oriented search types.

'Owner Type' Enhancements on Steps

The Seize, Release, Move, Produce, Consume and Search steps now have an additional 'SpecificObject' option for the 'Owner Type' property that that allows the owner that the step's action is in behalf of to be specified as a specific object reference. Before, the owner always had to be either the parent object containing the step or the associated object of the token executing the step.

This kind of feature provides more flexibility in process logic design. For example, an entity may go through a server, require a resource for processing and then the entity may wish to continue to another station without immediately releasing the resource. Process logic elsewhere in the model, regardless of whether the token executing the process is associated with the entity, can now easily command that entity to release the resource later in time.

Transfer and EndTransfer Steps 'Object Type'

The Transfer and the EndTransfer steps now provide an additional 'SpecificObject' option for the 'Object Type' property that that allows the entity being transferred to be specified as a specific entity object reference. Before, the entity always had to be either the parent object containing the step or the associated object of the token executing the step.

This kind of feature provides more flexibility in process logic design.

Travel Step 'Token Wait Action'

The Travel step now includes a 'Token Wait Option' property that can be set to either 'None (Continue)' or 'WaitUntilTravelCompleted'. This kind of feature provides more flexibility in process logic design, by allowing additional control of a token's flow through process logic.

Properties Enhancement

When a property for a model is specified, under the Appearance section of properties, the Display Name and Category Name may be specified to group the various properties. You are also now able to specify a

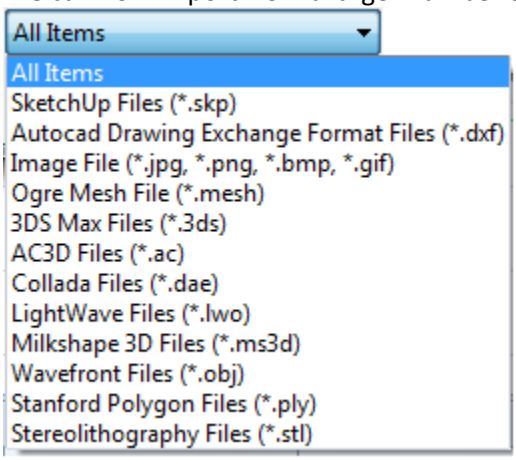
'Parent Property Name' such that one property is listed underneath another related property and accessible using the '+' on the parent property.

Shadows Change in Visibility Ribbon

The Shadows icon within the Visibility ribbon has now become two separate icons, including Direct Shadows and Diffuse Shadows. The **Direct Shadows** icon shows or hides shadows from a directional light source, like the sun. The **Diffuse Shadows** icon shows or hides soft shadows from other objects interacting with the illumination of an object. Only one of the shadows icons may be on at a time.

Expanded List of 3D Import Formats

We can now import from a larger number of 3D import formats, including the following:



Simio Enterprise Edition – Gantt Changes

Within the Simio Enterprise Edition, there have been several changes to make the entity and resource Gantt charts easier to use. When an object has 'Resource Logging' set as True, two additional properties are available. The Display Category property allows for hierarchical arrangement of resources within the Resource Plan Gantt (for example to display all resources of a given type or department grouped together). The Display Color property allows for user specified colors within the Resource Plan Gantt.

Additionally, if an additional column is added to the Resource Usage Log window for grouping (previous versions had this feature for Resource Gantt grouping), this column may be referenced within the Entity Workflow Gantt. This allows the Gantt chart for entities to be 'grouped' into various categories, as specified via the resource usage log (which typically may reference a data table column). See the Help page titled Resource Usage Log (Enterprise) for an example.

Flow Library Tank Object Enhancement

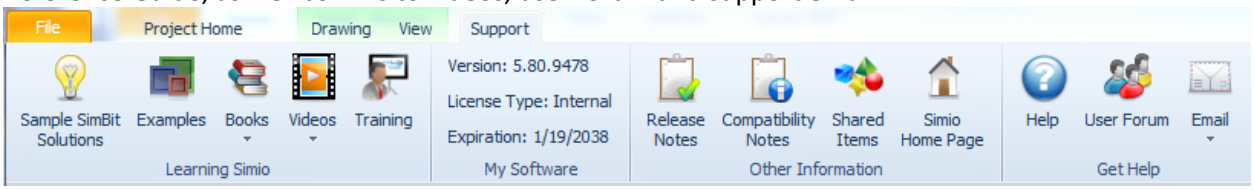
The Tank object within the Flow Library now has a 'State Assignments' section of properties to allow you to specify state assignments when the tank gets full or empty or its level rises or falls below a tank mark without having to go to add-on processes to do so.

Simio Release 5 – Sprint 80-81 – December 6, 2012

We are happy to announce the release of Simio Version 5. If you have been keeping up with interim (mostly internal) sprints, you will find that the most recent set of features is focused on many animation, GUI, and ease-of use features like the support ribbon. If you haven't seen Simio since the last major update (version 4) then you can look forward to reading the following 50 pages of release notes to answer the question "What's new?"

New Support Ribbon

A new ribbon has been added to the product that provides various support features that were previously found on the Start Page, in addition to numerous other resources for your use. This includes access to SimBits, Examples, and Books such as Introduction to Simio, Reference Guide and API Reference Guide, as well as links to videos, user forum and support email.

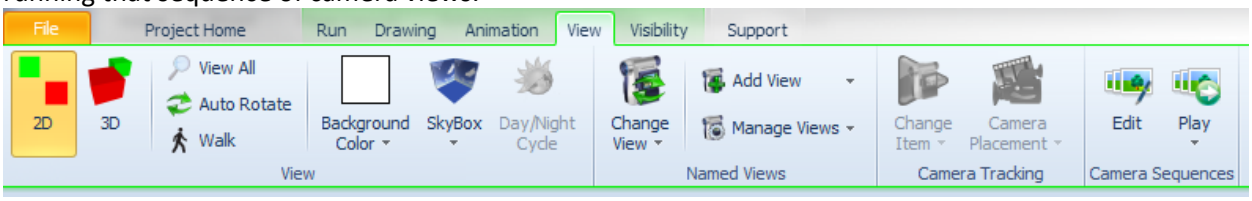


Additionally, information regarding customer license type and version number and maintenance expiration can be found on this ribbon. We are sure you will find this product information useful. When your maintenance or your product license itself is close to expiring, we now also provide a reminder message on the bottom status bar to help you avoid inconvenient expirations.

Camera Sequences and HotKeys on View Ribbon

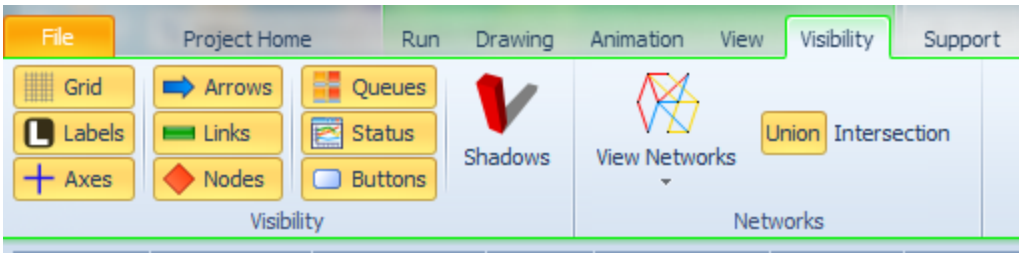
Two new buttons have been added to the View ribbon, including **Edit** and **Play** under **Camera Sequences**. A camera sequence is a sequence of instructions that can be played back either by selecting the sequence using the Play button or using a hotkey associated with the sequence. Named views, as well as the seconds to stay at that view and the amount of time to change views are specified for a given sequence. This sequence can then be used during the simulation run to demonstrate various views of the animation. The AirportTerminal example model contains an example of the new Camera Sequence. Simply click the Play button on the View Ribbon to see the camera sequences animated.

Both Named Views and the new Sequences support HotKeys. When adding/editing a named view or a camera sequence, you can now enter a HotKey as well as the Name. When a Named View has a HotKey assigned to it, hitting that key while in the Facility window, will take you to that Named View. When a Camera Sequence has a HotKey assigned to it, hitting that key while in the Facility window, will start running that sequence of camera views.



New Visibility Ribbon

With the addition of the above Camera Sequences buttons, we have moved some of the functionality from the View ribbon to a new Visibility ribbon. The ability to toggle on/off the visibility of the various items, such as grid, labels, arrows, nodes, etc., as well as network functions, have been moved to the new Visibility ribbon, as seen below.



F11 – Full Screen Viewing

The function key, F11, now allows the capability of collapsing all of the ribbons and other windows so that the simulation animation can be seen on the full screen. Pressing theF11 or Esc key while in full screen will then put back the ribbons and windows that were collapsed. This key can also be used in other windows, such as Processes, Data (for tables), Results, and Experiments.

Video Recording

Two new buttons have been added to the View ribbon to permit generation of video (.avi) files directly from within Simio. The Video group contains the Record Run Animation and Stop Recording buttons. Clicking on the Record Run Animation button will prompt the user to choose a name and location for their .avi file. The user will also be prompted to select a video compressor.

This capability can be used to generate simple videos of your animation files, more complex "how to" videos involving model-building, or even interaction with other applications. It is particularly useful when combined with the Camera Sequences option described above.

New Textures and Symbols

Six additional skins (textures) were added for more choices in representing rocks and minerals. You can also add your own skins to Simio simply by copying a jpg or png file into your Skins Simio folder (typically C:\Users\Public\Documents\Simio\Skins) and restarting Simio.

Many new symbols were added to the built-in library including new machines, various types of containers and storage places, and additional vehicles for hauling bulk materials and liquids.

Travel Step Enhancement

The Travel step now includes *Acceleration* and *Deceleration* properties for free space travel. This enables users to specify the rate at which an entity will gain speed going from a slower movement rate to a faster movement rate, as well as the rate at which the entity will lose speed going from a faster movement rate to a slower movement rate.

Transfer Step Enhancement

The Transfer step now has an additional property when transferring an entity into FreeSpace. This includes the *Facility Name*, which is the object whose facility free space into which the entity is to be transferred.

Lock Edits Enhancement

All items within the Facility window are now able to be right clicked and have 'Lock Edits' turned on/off. This will enable you to lock any object into place so that it does not move when navigating around the Facility window.

Ability to Duplicate Experiments

We have added the ability to duplicate entire experiments, which will duplicate the controls and data within the experiment but contain no results. Duplicating an experiment can be done in two ways. Through the Navigation window, the right click menu now allows you to duplicate, rename, or delete an experiment, as well as change experiment properties. Alternatively, in the Project window (by selecting the Project Name in the Navigation window and selecting the Experiment panel), right click now includes the ability to duplicate the experiment.

Option to Change Object References in a Model

When right clicking on a model within the Navigation window, the ability to change the model's object references is now available. This will open up a dialog displaying all the internal references the Model currently holds. From here, if a user placed two things down of the same name, they can change the name used to reference one in that particular model. They cannot change the reference names of objects in the same project, only things brought in externally (or objects that were in the project, but then they deleted).

New Libraries on the Simio User Forum

Like all Simio-provided libraries, these libraries are open so you can look at how they were built, learn from them, and subclass your own objects to improve them. But unlike our built-in libraries, these are not fully supported (although we can provide some help) and we may not continue enhancing them, particularly if we decide to build this capability into the base Simio products. We are anxious to receive your feedback on both libraries.

You can find these libraries via the **Shared Items** button on the Support Ribbon or in the Simio User Forum at http://www.simio.com/forums/viewforum.php?f=36

We encourage you to post your own contributions in this forum.

Extended Flow Library

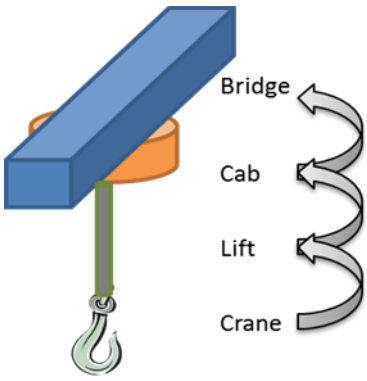
We have added an Extended Flow library and examples to the Simio User Forum. (http://www.simio.com/forums/viewforum.php?f=36). This library includes objects such as Solidifier, Liquefier, Filler, Extractor and Stockpile. A Solidifier can be used to convert a specified quantity of flow into discrete entities. A Liquefier converts a discrete entity into an outbound flow. The Filler is similar to a Combiner that combines input flow of a specified volume with a discrete entity and the combined entity departs the Filler. The Extractor is the flow equivalent of the Separator. It will separate a batch member entity from the parent entity, with the flow transfer of the batch member entity then discrete transfer of the parent entity once flow is complete. The Stockpile object is similar to a Tank with different animation graphics.

As we receive customer feedback on both the Flow Library and the Extended Flow Library we will probably build more of this type of capability directly into the Flow Library. But in the interim, hopefully this extended library will help in your modeling.

Crane Library

The Simio Crane Library is a collection of objects designed for modeling multiple cranes operating simultaneously in a bay. The library is provided as an example of complex material handling using the standard features of Simio. The Crane library may be used in conjunction with the Simio Standard Library, and Crane pickups are done using the standard TransferNode (e.g. the output side of a Server). Crane drop-offs can be done at either a BasicNode or TransferNode. The Crane Library can also be used with custom libraries as long as they support rider pickups using the standard transporter ride features.

The Crane Library consists of objects representing the **Bay**, **Bridge**, **Cab**, **Lift**, and **Crane** (the end effecter that actually picks up and drops off the item). These objects are combined together to model multiple cranes moving in a single bay. The separation of objects in this way allows you to use your own symbol for each component. A Crane movement occurs by first rising up from the pickup node to a specified travel height, traveling laterally at that height, and then lowering down to the specified drop-off node. All travel is done through free space without the need to explicitly draw a network. The Crane library also fully supports independent acceleration/deceleration and the ability for one crane to cause another blocking crane to move out of the way.



The initial posting of the crane library has a few known issues and additional features already under development. We recommend deferring production use until these issues are addressed (January 2013), but we encourage early adopter use and feedback to help us identify features and other areas where we can improve the library.

Simio Release 4 – Sprint 79 – October 29, 2012

This sprint includes the official 'Beta' release of the Flow Library. We've made some additional enhancements to the objects and now also have a 6 model SimBit, FlowConcepts.spfx included in the software. Several user-requested enhancements have also been made to this sprint, including a method for clearing statistics, an expansion of table searching capability and an option for orienting entities in free space.

New ClearStatistics Step

A new ClearStatistics step has been added to the Processes window. This step allows users to clear all model statistics or specific object/element statistics. Possible uses for this step include wanting to clear all statistics during an interactive run based on some triggering event or 'warmup' time, or wanting to collect and report periodic statistics (e.g., some logic that every hour writes out some object or element statistics and then clears them).

Search Step Enhancement

The Search step now has a new property within the Advanced Options section that allows you to *Search Related Rows Only* (True/False). If the searched table has a primary key / foreign key relationship, this property indicates whether or not to include only related rows in the search. If this property is 'False', then the search will include all rows in the table, regardless of table relationships. The previous sprint behavior only searched those related rows.

Travel Step Enhancement for Free Space Movement

The Travel step for free space travel now includes the option for auto orienting the moving object towards its destination. This option is used to automatically orient the entity such that its front is facing the travel destination. Otherwise, the entity will stay in its current orientation when initializing the travel movement.

Stacked Bar Animation Enhancements

The Stacked Bar on the Animation ribbon has been enhanced to include 'Floor Display' options that will display tank contents on the floor so users can see them in 2D. A gauge like display can be shown on the left or right side of the tank, if desired.

Flow Related Changes

FlowNode Object and Regulator Element – Initial Output Entity Type and Function

The FlowNode from the Flow Library (as well as the related Regulator Element) now have a function for 'FlowNode.FlowRegulator.CurrentDesiredOutputEntityType' (or for the Regulator element, 'Regulator.CurrentDesiredOutputEntityType') that allows dynamic changing of the entity type when leaving the node (or regulator). The related property name has been changed to be the *Initial Output Entity Type*.

FlowNode Object – Dynamic Updating for Maximum Flow Rate and Yield Factor

The FlowNode now includes a Dynamic Updating section of properties. These properties include the *Maximum Flow Rate Equation* and the *Output Yield Factor Equation*. These two properties allow optional dynamic updating of either maximum flow rate and/or the output yield factor of the node's flow regulator. The new *Update Interval* property allows you to specify the time interval of the updates.

New Flow SimBit

We have added a new SimBit to our SimBit library:

- **FlowConcepts.spfx –** This SimBit includes six (6) unique models that show various concepts using the Flow Library. These include the following:

- **SimpleFlow –** demonstrates flow from a FlowSource object, through a FlowConnector, to a FlowSink object.
- **FillingEmptyingTank** – demonstrates filling and emptying a tank object.
- **TransferringFromOneTankToAnother** – demonstrates transferring from one tank object to another.
- **MergedFlow** – demonstrates merging two different entity flows.
- **SplitFlow** – demonstrates splitting a single entity flow into two entity flows.
- **CreateDiscreteEntitiesBasedOnFlow** – demonstrates the use of monitors and events to create discrete entities when a tank reaches a certain flow level.

Simio Release 4 – Sprint 78 – October 12, 2012

The long anticipated "official" Flow Library is now available! We've also enhanced the experiment window including adding drop-down lists for non-numeric fields, such as states, tables, and object references.

Flow Library

Unless you are using Express Edition, which is limited to using only the Standard Library, you will now see a second library option immediately beneath the Library objects. This is a beta version of Flow Library Version 1.0. We are still fine tuning it (e.g., object icons and symbols are still in process), but at this point it is stable and ready for use. We think you will find that this library establishes an excellent foundation & framework for fluid/mass flow based modeling. And as you can see in the brief description below, there is a great deal of power and flexibility built into those objects. We will be adding additional objects based on designs already in early planning stage as well as future customer requests. Please give us your Flow Library feedback to help meet your needs.

Initial Set of Flow Library Objects

FlowSource – The FlowSource object may be used to generate a flow of fluid or other mass of a specified entity type. The source of supply is by default considered to be infinite, but alternatively may be configured to run out if a stopping condition is met.
The flow rate out of the source is regulated by its 'Output' flow node.

FlowSink – The FlowSink object may be used to destroy the flow of entities representing quantities of fluids or other mass that have finished processing in the model.
The flow rate into the sink is regulated by its 'Input' flow node.

Tank – The tank object may be used to model a volume or weight capacity constrained location for holding entities representing quantities of fluids or other mass. Features included in this initial design include:

- The ability to specify either a volumetric or weight based capacity for the tank.
- The ability to specify initial contents in the tank of (possibly) varying entity types.
- An *Auto Refill Mode* that allows you to easily specify logic for automatically refilling the tank when it becomes empty, when it contents level has become low, or when some other specified event occurs.
- 'Input' and 'Output' flow nodes to easily connect mass or fluid flow connections into or out of the tank. The flow rates into and out of the tank are regulated by these nodes.
- The ability to specify 'Low', 'Mid', and/or 'High' tank level marks, and optional add-on process triggers to perform some action when the tank level rises above or falls below a mark.
- Optional add-on process triggers to perform some action when the tank becomes full or empty.
- A *FlowContainer.Contents.Volume* level variable that may be monitored or referenced to return the current volume level in the tank.
- A *FlowContainer.Contents.Weight* level variable that may be monitored or referenced to return the current weight level in the tank.
- A *FlowContainer.CurrentVolumeFlowIn/Out.Rate* and *FlowContainer.CurrentWeightFlowIn/Out.Rate* state variables that may be monitored or referenced to return the current total volumetric or weight based flow rates into or out of the tank.

- A default 2D & 3D level animation of the volume or weight contents in the tank.

FlowNode - The FlowNode object is a node specifically designed to regulate the flow of entities representing quantities of fluid or other mass. Common applications of a FlowNode object include use as an 'Input' or 'Output' node for controlling flow into or out of another object (such as the FlowSource, FlowSink, or Tank objects). A FlowNode may also be used to model a flow control point in a network of links. Features included in this initial design include:

- The ability to specify whether the maximum flow rate for the flow node's regulator is volumetric or weight based.
- The ability to specify the initial maximum flow rate for the flow node's regulator. You can then dynamically adjust that maximum flow rate for process control logic purposes by assigning the node's *FlowRegulator.CurrentMaximumFlowRate* state variable a new value.
- The ability to completely enable/disable a flow node's regulator, equivalent to turning a 'pump' at the node on or off. A disabled regulator will not output flow regardless of its current maximum flow rate setting (i.e., regardless of how open the 'valve' is). Disabling/enabling a flow node's regulator may be done by assigning the node's *FlowRegulator.Enabled* state variable a new value.
- The ability to specify an *Output Yield Factor* for the flow node's regulator. This factor is entered as a ratio of outflow to inflow, and may be used to scale the flow into the regulator such that there is a physical loss or gain represented in the regulator's output flow. You may specify an initial output yield factor as well as dynamically change the output yield by assigning the node's *FlowRegulator.CurrentOutputYieldFactor* state variable a new value.
- The ability to specify an *Output Flow Entity Type* for the outflow produced by the node's flow regulator. **Development Note**: This feature in Sprint 79 is already planned to be changed to *Initial Desired Output Entity Type*. And the flow node will then provide a *FlowRegulator.CurrentDesiredOutputEntityType* state variable that can be dynamically assigned a new value.
- Provides *FlowRegulator.CurrentVolumeFlowIn/Out.Rate* and *FlowRegulator.CurrentWeightFlowIn/Out.Rate* state variables that may be monitored or referenced to return the current total volumetric or weight based flow rates into or out of the node's flow regulator.
- Input Flow Control: Both 'Single Flow' and 'Merge Flow' control modes are provided.
    - The 'Single Flow' control mode allows you to process inflows only one inbound link at a time, and prioritize those inflows either by inbound link weight or by flow entity priority. There is also a *Switch Control Variable* option that easily allows you to define a discrete variable for use as a switch control mechanism to dynamically change the current inbound link selection (e.g., to manually 'switch' from using one inbound link to using another inbound link).
    - The 'Merge Flow' control mode allows you to merge flow by only matching inflows that have the same entity types or by merging all inflows regardless of entity type. There is also a *Merge Allocation Rule* that provides a variety of options for how competing inflows into the flow node should be merged. Choices currently include merging flow proportionally based on inflow rates, proportionally based on fixed link weighted percentages (blending), or merging using a preferred order strategy where some inbound links have higher priority for merging than others.

- Output Flow Control: Both 'Single Flow' and 'Split Flow' control modes are provided.

    - The 'Single Flow' control mode allows you to send outflow to outbound links one link at a time by either outbound link weight priority or by shortest path to an assigned destination. There is also a *Switch Control Variable* option that easily allows you to define a discrete variable for use as a switch control mechanism to dynamically change the current outbound link selection (e.g., to manually 'switch' from using one outbound link to using another outbound link).
    - The 'Split Flow' control mode allows you to split flow using a *Split Allocation Rule* with a variety of options. Currently only two options are provided to either split flow proportionally (fixed percentages) based on outbound link weights, or to split flow using a preferred order strategy based on outbound link weights (e.g., send flow first to a primary line but if any overflow then send that to a secondary line).

FlowConnector – The FlowConnector object may be used to define a direct, zero travel distance connection from one flow node location to another. A *Selection Weight* property is provided to define priority or percentage based information for the connector, and will be used if the connector is attached to flow node with an input or output flow control mode requiring such link weight information.
*General Note – The links within the Standard Object Library (such as Paths and Conveyors) can also be used somewhat with flow nodes, but if trying to use a Path be sure to set the "Allow Passing" property to "False". Otherwise, the model run speed will likely be very poor.

Varying Shape for Flow Links

We have added the elliptical and trapezoidal shape for all links, such as Paths, Conveyors, and Connectors. These shapes affect only the flow type entities and the logical flow calculations (for example, an elliptical shaped link of width 1 and height 1 will carry less than a rectangular one). For trapezoidal shapes, an additional property, Trapezoid Ratio, is available. This value is the ratio of the top of the trapezoid to its bottom (which always set to the width of the Link). It can be any non-infinite value >= 0. A value of zero makes a triangle, a value of 1 makes a rectangle.

The type of shape will affect not only the flow calculations, but the shape of the available area graphic for the link, and the shape of the entities that move through the link.

Experiment Enhancements

In order to enable additional functionality, we have changed to a new technology underlying the experiment design window. You might notice some subtle changes.

- A new checkbox has been added in the heading of the first column. This permits easily toggling the active state on and off of all of the scenarios or any selected subset of scenarios. This is particularly useful when you want to only show a subset of the scenarios in the Response results window.
- Column sorting has been moved to a right click menu to allow column selection without changing sort order.
- The controls within the Experiment are based on the properties in a given model. Because these can be of various types (i.e., Tables, States, various Elements, Objects, etc.), these controls now display the drop-down lists available for each category from within the model. This will allow the user to much more easily select the control values for experiments.

- The responses within an experiment now include the Unit Type and Units for the response Expression. This allows users to display response results in varying display units. There are Units properties for the Upper Bound and Lower Bound of the response as well.

Simio Release 4 – Sprint 77 – September 18, 2012

We have added quite a number of new features to our Regulator element, in addition to adding statistics for Regulators, Containers and Links. These changes, along with continued development on our flow library objects, are moving us toward our release of the Flow Library at the end of the year. We've also made a significant enhancement to our experiments to allow for multi-objective optimization analysis – Simio is the first simulation software to feature this capability.

OptQuest Enhancements

When using the optional OptQuest for Simio add-in, the *Objective Type* property is now available to select either 'Single Objective', 'Multi-Objective Weighted' or 'Pattern Frontier'. This provides significant new flexibility in optimization.
Single Objective – OptQuest determines the set of control values that optimizes the experiment's Primary Response. This option is how OptQuest has worked in past sprints.
Multi-Objective Weighted – OptQuest optimizes across all responses with Objective set to Minimize or Maximize, taking into account each response's Weight value, to determine a single 'optimal' solution.
Pattern Frontier – OptQuest optimizes across all responses with Objective set to Minimize or Maximize, and finds the set of scenarios that are optimal, rather than a single 'optimal' solution based on weights.
* Note that the HospitalEmergencyDepartment example has been updated to incorporate the new multi-objective analysis using OptQuest.

New Entity Function
CurrentConnectorLink - If the entity object is currently using one or more connector links to transfer between two node locations, then this function returns a reference to the forward most connector being used. Otherwise the Nothing keyword is returned.
Note that a connector link does not have a logical length and thus is not considered to be physically occupied by an entity that is using it. To get a reference to the link that an entity's leading edge is currently physically located on, use the 'CurrentLink' function.
Flow Enhancements

We have continued with our flow enhancements including the following:

Flow Statistics
Container – Flow In and Flow Out of a container are now displayed in the simulation results.
Regulator – Flow Out is reported if it is greater than 0, while Flow In is reported if it is not equal to Flow Out (yield factor).
Link – Flow In and Flow Out are reported if at least one of them is greater than zero.

Regulator Enhancements

Initial Output Yield Factor – This new property should be entered as a ratio of outflow to inflow and may be used to scale the flow into the regulator such that there is a physical loss or gain represented in the regulator's output flow.

Output Flow Entity Type – This property now applies to single or merged flow from the regulator and is used to specify the entity type of the outflow produced by the regulator. If unspecified, then the output flow entity type will be determined by the inflow entity types.
Merge Allocation Rule – The PropretionalBasedOnExpression rule has been added, along with an additional related property *Merge Proportion Expression* to return the desired merge proportion for a flow request in the regulator's flow request queue.

Simio Release 4 – Sprint 76 – August 29, 2012

In Sprint 76, we have made many changes to various parts of the software including functions for additional flexibility in network travel, enhancements to the stacked bar, regulator and container for flow processing and additions to the Simio Enterprise edition.

New Link Functions

StartingNode = Returns a reference to the node located at the drawn starting point of this link.
EndingNode = Returns a reference to the node located at the drawn ending point of this link.

Network Element Functions

Distance(fromNode,toNode) Returns the shortest path distance between two nodes using this network.
NextLink(fromNode, toNode) Returns a reference to the next link on the shortest path between two nodes using this network.
NextNode(fromNode, toNode) Returns a reference to the next node on the shortest path between two nodes using this network. If the destination node is returned, then the two nodes are directly connected by a link.
PathExists(fromNode, toNode) Returns 'True' if a followable travel path exists between two nodes using this network.

New Entity Function

LastLinkUsed - Returns the most recent link used by the entity to complete a travel or instant transfer (in the case of a connector link) into a new node location. The value returned by this function is updated whenever the entity transfers from the end of a link into a node, or transfers across a connector from one node into another node.

Animated Stacked Bar Enhancements

The Stacked Bar graphic on the Animation toolbar has been enhanced to allow various 'types' including Container, Queue and Expressions (prior version animated only Queue contents).

- **Container**: When displaying a stacked bar for a Container, only the container element name is required and the volume contents of the container are displayed. The total volume capacity of the container then defines the outside portion (Total Expression) of the stacked bar.
- **Queue**: When displaying the Queue, a Total Expression property has been added to describe the total space available in the graphic.
- **Expression**: When displaying a stacked bar for Expressions, a repeat group of Expression and Color are available to graphically show various expression values within the stacked bar. This could include various model states or expressions. A Total Expression can be used to describe the total space available in the graphic.

Flow Enhancements

We have continued with our flow enhancements including adding infinite flow rates on the Regulator element, outbound link rules, merging within Containers and merging/transfers to different destinations with the Regulator. These changes will allow us to continue developing our 'flow library' which is intended to be released at the end of the year.

Simio Enterprise Edition Enhancements

Within Simio Enterprise Edition, several changes have been made to allow for comparison of various model alternatives. After a user performs the 'Analyze Risk' within the Operational Planning ribbon, the

Results panel option on the left side allows for viewing of various results, including Target Summary, Target Detail, Risk Plots and others. Within the Target Detail tab, there are options on the Operational Planning ribbon for Plan Comparisons. These include 'Save for Comparison' and 'Show Differences' buttons.

The '**Save for Comparison**' button will save the current model (including logic, data tables, etc.) and place the saved information in a 'Baselines' folder in the Navigation window, named by the ModelName_Date_Time.

Once at least one baseline has been saved, the '**Show Differences**' button becomes available and acts as a toggle between the Target Detail report for the baseline and a Target Detail report that shows the 'Differences' or 'Changes from Baseline' between the two models. This allows users to compare various models and discover what difference, if any, the model changes make to the final target results.

Simio Release 4 – Sprint 74-75 – August 9, 2012

A major new feature in Sprint 75 is the addition of many enhancements for free space movement. This allows much more flexible free space travel for entities, vehicles and workers. We have also continued to work on Flow Processing enhancements, including the addition of flow states for Containers, Regulators and Links.

Free Space Movement Enhancements

Often you may want to travel between many locations, but don't want to manually interconnect every possible node combination. While using a network with shared links saves a lot of time, it can sometimes still be tedious. In the case where you are looking for straight-line movement between nodes, Simio now has a new solution: **Free Space Movements**. While Simio has always supported movement through free space, it has now been made much easier with the addition of the Travel Step, updates to the SetNode and SetNetwork steps, enhancements to the Entity object and the Node, Worker and Vehicle library objects, and a host of new functions. In addition this feature provides support for easier detailed free space movements such as those you may want to include in a custom object or library.

New Travel Step

A new Travel step has been added for use in the Processes window. This step may be used to do a direct, straight-line movement of an entity in free space to a specified location. The executing token will be held at the Travel step until the travel movement is completed. The new destination may be specified as a fixed object, absolute coordinates or relative coordinates to the entity's current location. A maximum desired movement rate and ending movement rate may also be specified.

Entity Objects (including ModelEntity, Worker, Vehicle)

The *Initial Network* property now provides the choice 'No Network (Free Space)' so that free space movement of entities, workers and vehicles may be incorporated. The default value of this property still remains 'Global'.

There are now new OnEnteredFreeSpace standard processes for the Worker, Vehicle and ModelEntity objects. Please see the Known Anomalies regarding information on free space travel for entities.

New Node Functions

- **IsInputNode** – Returns 'True' if this node is an external input node for entering another object. This function replaces the obsolete function, IsExternalInputNode.
- **IsOutputNode** – Returns 'True' if this node is an external output node for exiting another object.
- **Nearest.Node** – Returns the node that is the shortest straight-line distance from this node.
- **Nearest.InputNode** – Returns the input node that is the shortest straight-line distance from this node.
- **Nearest.OutputNode** – Returns the output node that is the shortest straight-line distance from this node.

Step Changes

- **SetNetwork step** - *New Network Name* property now provides the choice 'No Network (Free Space)'. This lets you dynamically change an entity or transporter from using a network to only traveling in free space and vice-versa.
- **SetNode step** – If setting a specific node name, the *Node Name* property now provides the choice 'No Destination' (similar to <null> in prior sprints).

Free Space SimBits Changes

We have updated our SimBits library with changes based on the addition of the free space travel enhancements listed above. Two SimBits, EntityMovementInFreeSpace.spfx and MovementInFreeSpaceWithATargetLocation.spfx have been deleted and replaced by:

- **FreeSpaceMovement.spfx –** This SimBit displays the use of various types of free space travel. There are three models within the project. The first demonstrates entity free space movement from a Source to multiple Servers then to a Sink. The second demonstrates entity free space movement between two nodes after using network based travel (including the new Travel step). The third model demonstrates both entity and vehicle free space movement.

Flow Enhancements

Simio has always provided support for discrete and continuous flow, but a set of changes to the Simio engine (e.g., the Steps, Elements, and other related low-level changes) are nearly complete that will set a new standard of modeling capability. Examples of using these new features illustrated with a sample library can be found on the Simio Insiders portion of the user forum, under the SI Material Handling topic: http://www.simio.com/forums/viewtopic.php?f=10&t=762. While these low-level features are nearly complete, we are anxious to hear your feedback at http://www.simio.com/forums/viewtopic.php?f=10&t=710 as we continue with our design and implementation of the "official" flow library that is expected to be completed late this year.

New Link States

- **CurrentVolumeFlowIn** – State used to get the current total volume flowed into a link.
- **CurrentVolumeFlowOut** – State used to get the current total volume flowed out of a link.
- **CurrentWeightFlowIn** – State used to get the current total weight flowed into a link.
- **CurrentWeightFlowOut** – State used to get the current total weight flowed out of a link.

New Regulator States

- **CurrentMaximumFlowRate** – State used to get or set the current maximum flow rate of a regulator.
- **CurrentVolumeFlowOut** – State used to get the current total volume flowed out of a regulator.
- **CurrentWeightFlowOut** – State used to get the current total weight flowed out of a regulator.

New Container States

- **CurrentVolumeCapacity** – State used to get or get the current volume capacity of a container.

- **CurrentWeightCapacity** – State used to get or set the current weight capacity of a container.
- **CurrentVolumeFlowIn** – State used to get the current total volume flowed into a container.
- **CurrentVolumeFlowOut** – State used to get the current total volume flowed out of a container.
- **CurrentWeightFlowIn** – State used to get the current total weight flowed into a container.
- **CurrentWeightFlowOut** – State used to get the current total weight flowed out of a container.

Splitting Flow

You can now build a model with a node's Output Flow Strategy set to Split Flow and using an appropriate rule to allocate the flow between outbound links.

Miscellaneous Changes

- Infinite Regulator flow rates have been added
- Connector links for flow (links with zero length) are now supported

Other Miscellaneous Changes

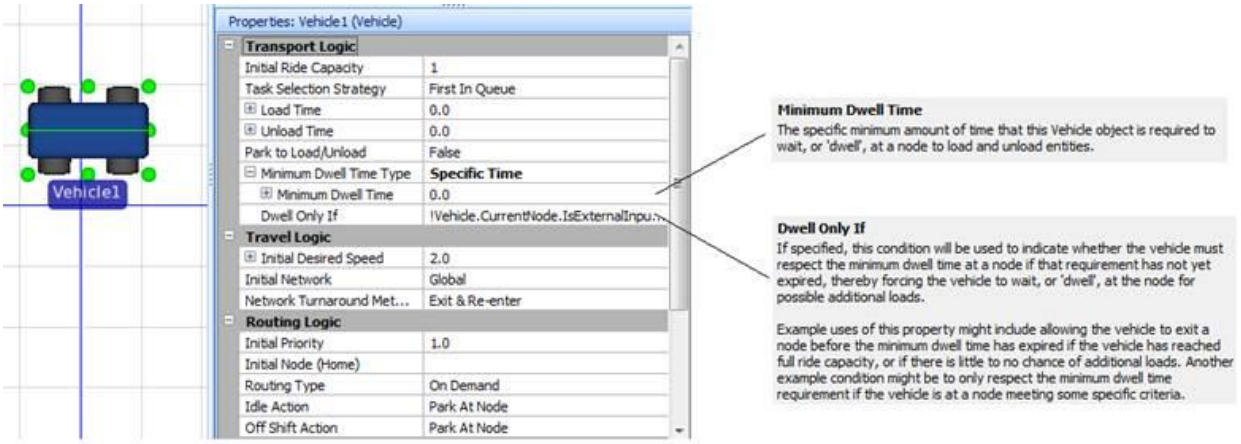
- Ctrl-Wheel now zooms within an individual process in the Processes window
- **AddRow step** – Now supports a 'SpecificObjectorElement' Object Type value, asking for the specific 'ObjectOrElement' if that option is specified.
- **Search step** – Improved trace when running the simulation model to include the exact items found by the search and the *Save Index Found* and *Save Number Found* values, if specified.

Simio Release 4 – Sprint 73 – June 26, 2012

The major new feature in Sprint 73 is an enhancement to Vehicle and Worker to easily support flexible Dwell Time. We also added new macros to support object development by more advanced users. We also made some subtle changes that improve overall product ease of use and robustness.

Vehicle & Worker

- **Minimum Dwell Time** – New capability to make it easier to control how long a vehicle will wait at a node before proceeding. The current default continues to be to wait only until all entities that can be immediately loaded are loaded. Now it is easy to add flexible waiting ("dwell time") options to do such things as wait until full, wait until a scheduled departure time, or wait for a specific condition or event.



Property Default Values

Macros – There are 'macros' that can be put in the default value of a property that will expand to the correct value when an object when that property is instantiated. The new macros are:

- $(InstanceName)
- $(DefinitionName)
- $(AssociatedObjectInstanceName)
- $(AssociatedObjectDefinitionName)

For example previously if you renamed ModelEntity to something else, then placed an instance, you would get an error about ModelEntity.Picture not being a valid property value. The default value for the property is now set to "$(DefinitionName).Picture", which now resolves correctly when the instance is placed, and no errors occur. This feature will make it easier to design library object which can be reused without errors like this.

Miscellaneous

Zoom – Zoom now works better in that zoom in, then out will return you to the original zoom level. And a problem with some "fuzzy" graphics on view change has been corrected.

Process Window – Categories now have a better look and feel as well as some small bug fixes.

Simio Release 4 – Sprint 72 – June 1, 2012

In this sprint, we have added many new entity functions in support of sequence planning and scheduling. These functions can be used to access various pieces of information within an entity's sequence table/plan. Additionally, we are continuing our development on flow processing. A new example library is now available on our user forum for a preliminary look at the various features within flow.

New Entity Property

- **Due Date Expression** – The expression used to return a ‘due date’ value for an entity of this type.

New Entity Sequence Related Functions

- **Entity.Sequence.NumberJobStepsRemaining** – Returns the current number of steps remaining in the entity’s assigned sequence, including the current job step.
- **Entity.Sequence.ExpectedOperationTimeRemaining** – Returns the sum of the expected operation times for the remaining steps in the entity’s assigned sequence, including the current job step.
- **Entity.Sequence.CriticalRatio** – Returns an index for determining how much the entity’s assigned sequence is considered to be on schedule given the due date and expected operation time remaining. A value of 1.0 is ‘on schedule’. A value less than 1.0 is behind, while a value larger than 1.0 is ahead of schedule. This function returns a value equivalent to the expression ‘(Entiyt.Sequence.DueDate – Run.TimeNow) / Entity.Sequence.ExpectedOperationTimeRemaining’.
- **Entity.Sequence.SlackTime –** Returns the expected amount of time that would be left until due date after completing the entity's assigned sequence, if the sequence's remaining operation time was started now. This function returns a value equivalent to the expression '(Entity.Sequence.DueDate – Run.TimeNow) - Entity.Sequence.ExpectedOperationTimeRemaining'.
- **Entity.Sequence.SlackTimePerOperation -** Returns the average slack time per remaining operation for the entity's assigned sequence. This function returns a value equivalent to the expression 'Entity.Sequence.SlackTime / Entity.Sequence.NumberJobStepsRemaining'.
- **Entity.Sequence.DueDate -** Returns a due date for the entity to complete its assigned sequence. The 'Due Date Expression' property for the entity type is used to get the due date value.
- **Entity.Sequence.ModifiedDueDate -** Returns a modified due date value for the entity's assigned sequence, which is the highest of either the due date or the expected completion date if the sequence's remaining operation time was started now. This function returns a value equivalent to the expression 'Math.Max(Entity.Sequence.DueDate, Run.TimeNow + Entity.Sequence.ExpectedOperationTimeRemaining)'.
- **Entity.Sequence.DestinationNodes -** The destination nodes in the entity's assigned sequence.
- **Entity.Sequence.DestinationNodes.NumberItems -** Returns the number of destination nodes in the entity's assigned sequence.
- **Entity.Sequence.DestinationNodes.FirstItem -** Returns a reference to the first destination node in the entity's assigned sequence.
- **Entity.Sequence.DestinationNodes.LastItem -** Returns a reference to the last destination node in the entity's assigned sequence.
- **Entity.Sequence.DestinationNodes.ItemAtIndex(index) -** Returns a reference to the destination node at a specified step index position (or job step) in the entity's assigned sequence.
- **Entity.Sequence.DestinationNodes.IndexOfItem(node) -** Returns the one-based step index (or job step) of the first occurrence of a specified destination node in the entity's assigned sequence. If the entity does not have an assigned sequence then the value 0 is returned.

New Node Property

- **Sequence Expected Operation Time** – The expression used to estimate an expected operation time if this node is in the destination sequence of an entity's assigned sequence table.

Simio Release 4 – Sprint 71 – May 20, 2012

As in the past several sprints (which have not been released externally), we are continuing to work on features to support flow in future sprints. The Container and Regulator elements, as well as changes to the Transfer step, allow basic flow type objects to be designed and tested. We've also continued to make enhancements to our standard product, such as adding transporter functions and enhancing the Wait step.

New Transporter Functions

- **RideStationLoad** – Returns the current 'load' on the station location that holds the entities riding on the transporter. The ride station 'load' is defined as the sum of current entities reserving space on and intending to ride the transporter plus the current number of riders.
- **RideStationOverload** – Returns the current difference between the load and capacity values (a positive difference indicating an 'overload') for the station location that holds entities riding on the transporter. The ride station 'load' is defined as the sum of the current entities reserving a spot on and intending to ride the transporter plus the current number of riders. This function returns the difference between the load and the current ride capacity (overload = load – capacity).

Revised Entity Function

- **Entity.ReservedTransporter** –This function is now set to the reserved transporter until the entity is actually loaded and has moved into the transporter's RideStation. (Previously, this value was set to 'Nothing' during any loading time for the entity to move onto the vehicle). Thus, if an entity has reserved space on and intending to ride a specific transporter (but has not yet loaded onto it), its ReservedTransporter function is set to that transporter. The Entity.CurrentTransporter function is used once the entity is riding on the transporter.

Standard Library Changes

- **Vehicle/Worker** – The Ride Capacity property is now an expression property instead of an integer property. This allows greater flexibility in specifying the capacity for vehicles and workers. This property was also renamed *Initial Ride Capacity*.
- **TransferNode** – The transporter 'Selection Expression' default value is now 'Candidate.Transporter.RideStationOverload' instead of 'Candidate.Transporter.RideCapacityRemaining'.

Routing Group Element / Transfer Node Enhancement

The Routing Group element now supports the Node List property being independently evaluated for each entity route request (per table entry, for example). Within the Transfer Node, for example, if selecting the entity destination using the 'Select From List' option, the name of the node list might come from table data and differ based on the entity.

Ride Step Changes

The transporter 'Selection Expression' default value is now 'Candidate.Transporter.RideStationOverload' instead of 'Candidate.Transporter.RideCapacityRemaining'

Wait Step Enhancement

The Wait step now allows multiple events and event conditions to be evaluated before releasing entities from the step. The user has the ability to specify for the tokens to wait until all events have been fired or wait until any one of the events have been fired before releasing the tokens from the Wait step.

New SimBit

We have added a new SimBit to our SimBit library:

- **HourlyStatistic.spfx –** This SimBit displays the use of the Timer and TallyStatistic elements in calculating the average number of entities in a given input buffer on an hourly basis.

Flow Enhancements

The Container and Regulator elements have been added and the Transfer step has been revised to support the addition of Flow type objects in the future. The Container element is used to define a volume or weight capacity constrained location for holding entities representing quantities of fluids or other mass. The Regulator element may be used to regulate flow transfers of entities into or out of a location. The Transfer step has been modified to include transfers to and from Containers.

It is important to note that while we currently have some new elements and revised steps to support flow, these features should not yet be used within any production models. Examples of using these new features within an object and/or small library are available upon request. We strongly encourage your feedback through the Simio Insiders portion of the forum, under the SI Material Handling / Updated Flow Design, http://www.simio.com/forums/viewtopic.php?f=10&t=710.

Simio Release 4 – Sprint 70 – April 20, 2012

Again in this sprint, much of our development time has been spent with flow related features. We've enhanced a few of our node related functions, as well as added capability to our Search window. Some changes have been made to the way negative costs are handled within our costing areas as well.

New Node Functions

- **AssociatedNode** (description modified) – For an external node, this function returns a reference to the object that the node is attached to and which may be entered or exited using the node. If the node is not an external node, then the Nothing keyword is returned.
- **AssociatedStation**– For an external input node, this function returns a reference to the immediate station location inside the node's associated object that may be entered using the node. If there is no station location, then the Nothing keyword is returned.
- **AssociatedStationLoad** –** For an external input node, this function returns the current 'load' on the station locations inside the node's associated object that may be entered using the node. The associated station 'load' is defined as the sum of current entities en route to the node intending to enter the stations, plus the current entities already arrived to the node but still waiting to enter the stations, plus the current entities occupying the stations.
- **AssociatedStationOverLoad** –** For an external input node, this function returns the current difference between the load and capacity values (a positive difference indicating an 'overload') for the station locations inside the node's associated object that may be entered using the node. Overload = Load – Capacity.
- **NumberTravelers.RoutingInToEnterAssociatedObject** – For an external input node, this function returns the current number of travelers with their destination set to this node and which can enter the node's associated object.

**Unlike the deprecated InputLocation.Load and InputLocation.Overload functions, the new AssociatedStationLoad and AssociatedStationOverload functions will consider the 'Processing' station contents of standard library objects such as Server, Combiner, Workstation, and Separator.

Costing Changes

The object and cost center element Cost state can now have a negative value or rate. Previously, Simio was turning negative values into 0.0.

Additionally, we have several built-in cost drivers including resource usage rate/idle rate/per use costs, station per use/holding rate cost, for making it easier to accrue costs to cost centers and object costs. Previously Simio was turning negative values into 0.0 for all the built-in cost drivers. We have added run-time error messages on all of our built-in cost parameters if the user enters a non-negative value.

Search Window Enhancements

We have enhanced the Search window (Project Home ribbon) to include object definition names in the searches. For example, if you have 10 Server type objects in your model, but none of the with the word 'server' in the name, previously, a search would not have found these. Now, the Item Type is shown a 'Server' and all server definitions within a Facility window will be located.

Simio Release 4 – Sprint 69 – April 5, 2012

In this sprint, much of our development time has been spent with flow related features. We've also added some user requested features, such as allowing property values within schedules and applying the external view symbol to a changed symbol of an object.

Animation Ribbon – Stacked Bar Graphic

We have added a new graphic for displaying information about queue contents called a Stacked Bar. The Stacked Bar graphically displays the value of some expression evaluated for each item in a designated queue as a display of stacked blocks (for example, ModelEntity.Volume for a Server1.Processing.Contents queue). This can be used in the Facility window or External View window of an object.

New SimBits

We have added two new SimBits to our SimBit library:

- **AppointmentArrivals.spfx –**This SimBit shows model arrivals that occur at a specific time, such as a scheduled appointment. In the example, entities arrive every 15 minutes between 8:00 – 4:30, but also have a time deviation to demonstrate early and late arrivals.
- **WorkersArriveLateToShift.spfx –**This SimBit demonstrates the use of schedules for a worker, but also considers the fact that sometimes the worker arrives late to his/her shift. This is done by using a 'sub-classed' Worker object and adding additional properties for the worker's probability of lateness and late time.

New Container and Regulator Elements

We have added two new elements to the Elements panel that will be used in flow processing, including a Container and a Regulator. These elements are still under development.

External View Symbol

We added an option in the Right-Click menu when selecting an object that will allow the user to revert back to the 'external view' symbol originally specified. For example, perhaps you place a Server and change its symbol to another graphic. Later, you may wish to change the symbol back to its original default symbol, as shown in the External View of the object.

Schedule 'Value' and 'Cost Multiplier' Enhancements

Within a Schedule, under the Day Pattern section, the Work Periods specified allow the specification of a 'Value' which is the capacity value for a given resource. Additionally, a 'Cost Multiplier' can be specified that will factor the cost associated with that given work period and resource. Both of these fields may now be specified as an expression, which may include a property value. This will allow both fields to be changed through experimentation as well.

It is important to note that 'Value' and 'Cost Multiplier' fields are also available within various Exceptions specified (in the Work Period Exceptions within the Schedule, as well as within the Work Period Exceptions specifically within the object's properties). These currently do NOT support properties at this time, but should in the next sprint.

New Node Function

- **AssociatedObject** – For an external node, returns a reference to the associated object that may be entered or exited using the node. (For example, for Input@Server1 basic node, this function would return Server1 as its associated object).

Planning Tab – New 'Logs' View (Enterprise Edition)

Within the Planning Tab of the Enterprise Edition, you will see that the views have changed slightly to include a new 'Logs' view. We have taken the various logs, including the Resource Usage Log, Resource State Log and Constraint Log, out of the Entity Workflow and Resource Gantt areas and into their own view. This will enable users to move more easily between the various log files, without having to move from Entity Workflow to Resource Gantt and back, as was the case with the previous layout.

Simio Release 4 – Sprint 68 – March 15, 2012

In this sprint, we have continued to add features to Simio to support Flow Processing in the near future. These changes include the addition of weight and volume states (and rates) to entities, as well as some node internal changes to prepare for flow support. Many of our other changes are customer related enhancements, including the addition of arrays for string, element reference and object reference states, as well as variable expression threshold monitoring. We are continuously adding flexibility to our Simio Enterprise edition, including new features for Gantt sorting and filtering.

Entity State Changes

Entities now have Volume and Weight states. The volume is referenced as ModelEntity.Volume and represents the current *logical* volume of the entity. If not defined, the entity's volume will be the *implicit* volume as calculated by the Size parameters. The weight is referenced as ModelEntity.Weight and has an initial value as calculated by the Density and Volume of the entity. Volume and Density can be specified within the Entity's property grid. Any changes to weight (or volume) will change the volume (or weight) to keep a constant density of the entity. The ratio of the Size implicit volume to the logical volume is kept constant during the run.
A 'Rate' parameter is also available on the Volume and Weight states, as well as on the Size state under each of its Length, Width, and Height parameters. The Movement state of the entity also has this similar 'Rate' parameter. These 'Rate' states can be monitored as continuous variables.
Based on the above additions of volume and weight with the entity, the Station element's Contents queue now has the following read-only parameters as well – Volume, Volume.Rate, Weight, Weight.Rate. These represent the cumulative volume and weight contents of the queue. These can be used in a Monitor, so that the total volume or weight of objects in a station doesn't exceed a given value, for example.

String and Element Reference State Arrays

The String, Element Reference and Object Reference states can now be defined as arrays. In the past, these states were scalar values.

New 'Before Processing' Add On Process Trigger

Within the Server, Separator, and Combiner, a new 'Before Processing' add-on process trigger has been provided. This trigger occurs when the entity has been allocated the server/combiner/separator capacity, but before entering (or ending transfer to) the object's processing station.

Path, Conveyor and TimePath capacity Changes

The Path, Conveyor and TimePath objects in the standard library have been modified to have an Initial Traveler Capacity that can be an expression (previous software was 'Traveler Capacity' and was a static integer). This allows the dynamic changing of the capacity by using the 'NodeName.CurrentTravelerCapacity' state.

Node Changes

A node object in Simio now provides the following property:

- **Initial Traveler Capacity –** The initial maximum number of travelers that may simultaneously occupy the node.
- **Entry Ranking Rule –** The rule used to rank entry into this node among competing travelers.
- **Entry Ranking Expression –** The expression used with a 'Smallest Value First' or 'Largest Value First' entry ranking rule.

Nodes also have the following states:

- **EntryQueue** – The queue of entities that are waiting to enter the node.
- **CurrentTravelerCapacity** – State used to get or set the current maximum number of travelers that may simulataneously occupy the node.

Nodes also have the following functions:

- **NumberTravelers** – Returns the current number of travelers occupying this node.
- **NumberTravelers.RoutingIn** – Returns the current number of travelers with their destination set to this node.

The standard library node objects, BasicNode and TransferNode, are no longer 'resources'; therefore, they no longer have the Capacity, Ranking Rule and Ranking Expression properties. The new properties listed above have replaced them. We're confident that the priority merging flow through a node is working better with this new approach.

Note about behavior change within the BasicNode/TransferNode: Though the traveler capacity functionality now provided by BasicNode & TransferNode is very similar to the resource-based approach, one difference is that now an entity cannot actually enter the node and execute the 'OnEntered' process and the 'Entered' add-on process trigger until it actually is allocated traveler capacity and thus selected from the node's EntryQueue. Before, an entity would always first enter the node, execute the 'Entered' add-on process trigger if specified, and then have to seize the node resource to continue 'crossing' through the node.

Auto-Completer and Drop-Down List Changes

We've taken non-resources off of 'Seize' drop-down lists and no longer show resource functions and states for non-resource objects in the auto-completer. This should make model building using lists easier to use and select from.

Monitor Element Enhancement

For Monitor elements that have a *Monitor Type* set to 'CrossingStateChange' , the *Threshold Value* has now changed to *Initial Threshold Value*. The monitor now has a state 'MonitorName.CurrentThresholdValue' that can be changed during the simulation run. Previously, this threshold was a constant real value. This provides additional flexibility for users to have changing thresholds for a given Monitor element.

Gantt Chart Enhancements (Enterprise Edition)

If you change row heights or expanded states within the Gantts, Simio now tracks this and remembers it across re-runs.

Gantt charts can now be filtered by resource or entity by right clicking on the resource or entity and selecting the Filter to option. This will filter both the Gantt as well as the associated Log. This is available within both the Resource Plan Gantt, as well as the Entity Workflow Gantt.

Resource Usage Log Columns (Enterprise Edition)

The Resource Usage Log now has an option to add columns within the Log. These additional columns have the option of being displayed in either the Resource Gantt or the Entity Gantt. The column values are calculated based on an expression value. This is useful for showing additional related info in the Gantt charts.

For example, if you would like to sort the Resource Gantt in a particular manner, you can add a column to the Resource Usage Log that would return a sorting value (1, 2, 3, 4, etc.). This may be based on a table entry or any expression specified. The most recent values in the log for that resource are then shown in the Gantt and can be sorted on.

Simio Release 4 – Sprint 67 – February 17, 2012

In this sprint, we've continued to enhance the output reports and charts within Simio Enterprise Edition. Additionally, object referencing flexibility has been increased by providing more capability of how various functions are referenced. We've added multiple unit measures in anticipation of additional Simio features in future sprints.

Dynamic Objects within Resource Plan – Gantt (Enterprise Edition)

The Gantt chart within the Resource Plan panel of the Planning tab for Simio Enterprise has been enhanced to now include dynamic objects, such as Workers and Vehicles. The Gantt chart for these objects shows the order of entity processing , any constrained entities and resource state by individual unit within the population (i.e., Worker[1] and Worker[2] if the object named Worker has an *Initial Number in System* of '2').

States and Targets Displayed in Properties Window (Enterprise Edition)

The States and Target data that is generated for a given plan is now displayed within the Properties window for a given Entity using a particular Resource. This provides easier viewing of information about states and targets within a table in a grid format.

Object Referencing

We have added the ability to get a reference to an object's external input or external output node, which will allow the user to get property, state or function information on an object's external nodes if they have a reference to the object itself. For example, if you have a Server and you want to get a reference to its external input node, you could type 'Server1.Input.NumberRoutingIn' to determine the number en route to that input node. Alternatively, if you have an ObjectReferenceState that is holding a reference to Workstation1, then you could reference the queue for the output node by specifying 'ObjectReferencState1.Workstation.Output.RidePickupQueue.NumberWaiting'.

The keyword 'Is' can now be used with an ObjectInstanceProperty so the user can decide if an object is the same type as the object specified in an object's ObjectInstanceProperty. As an example, if you have an object, let's call it 'submodel', and you put an ObjectInstanceProperty on that object, then you can now say Is.ObjectReferenceProperty in your logic. So, you might have an instance of Submodel in the Model and the ObjectInstanceProperty is set to 'PartA' but it has PartA and PartB going into it. The logic might contain a Decide Step that asks 'Is.ObjectInstanceProperty', which is essentially asking 'Is.PartA' but only because that instance's ObjectInstanceProperty is set to 'PartA'.

Units Settings

We've added a Units Settings button with pull down to set the units to display times, lengths, volumes, mass, travel rates, volume flow and mass flow. This button can be on many ribbons and affects the units displayed within the reports, pivot grid, watch window, status bar and trace window.

Simio Release 4 – Sprint 66 – January 30, 2012

In this sprint, we've added more flexibility within our Standard Library objects, as well as made some enhancements for our new Enterprise edition software. Work schedules have been enhanced to include a cost multiplier (for overtime, as an example), and have also been added to the Vehicle object. User requested enhancements include the addition of new functions for entities and a new EndRun step, which allows the user to dynamically end the simulation run. We've started adding Enterprise edition specific SimBits to the software as well.

Work Schedules for Vehicles

Vehicles have been enhanced to include Work Schedules, similar to Workers and other Standard Library objects. The Vehicle now includes an 'Off Shift Action' for determining its behavior when it goes offshift. Additionally, two add-on process triggers, Off Shift and On Shift have been added for users to modify logic when a vehicle goes on or off shift.

New EndRun Step

A new EndRun step has been added for use in the Processes window. The EndRun step may be used to set the ending time of the simulation run to the current time. This will cause the run to end once all simulation events scheduled for the current time have been processed.

Shadows within the View Ribbon

The Shadows icon within the View ribbon controls whether or not the shadows created by each object in the model are displayed. The light source in Simio is fixed in the corner at a 45 degree angle. However, if shadows are used in conjunction with a dynamic Skybox and the Day/Night cycle, the shadows will be adjusted for the change in the light source. Shadows were also added to the ThemePark example.

Functions for Entities

- **Sequence.CurrentJobStep** – Returns the current row index into the entity's assigned sequence table.
- **Sequence.NumberJobsteps** – Returns the total number of rows (i.e., job steps) in the entity's assigned sequence table.

Cost Multiplier support for Schedules

A *Cost Multiplier* property has been added to Work Schedules within the Day Patterns definition area. This multiplier is applied to the idle and usage cost rates of a resource following the work schedule specified. The *Cost Multiplier* property can also be found within the Work Period Exceptions when defining a GeneralException.

New SimBit

We have added a new SimBit to our SimBit library:

- **UsingAddRowAndOutputTable_Enterprise.spfx** demonstrates how to use the AddRow step in conjunction with Output tables. This SimBit applies to Simio Enterprise edition, which includes Output tables. Simulation data is written out to a table at the end of a simulation run, including entity entry and exit times from the system, as well as which server the entity has been processed on.

Replication Runner (Enterprise and Team Editions)

Simio Replication Runner is a program that allows an Experiment in Simio to distribute its replications to other computers on the local network. When other network computers are running Simio Replication Runner, the Simio experiment will use the resources on that computer for the experiment run, thus distributing the processing requirements across multiple computers on the network.

Resource Plan Gantt Chart Changes (Enterprise Edition)

Within the Resource Plan Gantt chart within the Enterprise edition, we've modified the presentation of Resource States, as well as Exceptions to be easier to view as well as edit.

Additionally, when a specific entity at a given resource is highlighted within the Resource Plan's Gantt chart, the table properties of that entity are now displayed within the properties window on the right. Any editable properties can then be changed directly within the Gantt view without changing to the Data Tables window.

Simio Release 4 – Sprint 65 – January 11, 2012

In this sprint we continue to enhance features within our product based on user feedback. Some of this sprint's changes include usage time based failures for resource, vehicle enhancements, new SimBits, and a number of new functions that have been added to provide the user with greater modeling flexibility and use of object references.

Usage Based Failures for Resources

Resources now have the usage time based failures within the Reliability section of properties. This provides greater functionality for resources, allowing downtimes to be based on not just calendar time, but alternatively based on the amount of time that the resource is actually in use.

HomeNode Variable for Worker / Vehicle

A new HomeNode variable has been added for users to dynamically assign the home location for a given worker/vehicle, using either Vehicle.HomeNode or VehicleName[x].HomeNode. The Initial Home location of both workers and vehicles can be changed during the simulation run using an Assign step.

Day/Night Icon for SkyBox

With dynamic skybox views in 3D, there is now a new day/night icon that can be turned off/on to display ambient light and a moving sun/moon.

Functions for Resource-enabled objects

- **Capacity.CurrentOwners** – The objects that currently own (have seized) capacity units of this resource.
- **Capacity.CurrentOwners.NumberItems** – Returns the number of objects that currently own (have seized) capacity units of this resource.
- **Capacity.CurrentOwners.FirstItem** - Returns a reference to the first owner in the list of owners that have currently seized capacity units of this resource.
- **Capacity.CurrentOwners.LastItem** - Returns a reference to the last owner in the list of owners that have currently seized capacity units of this resource.
- **Capacity.CurrentOwners.ItemAtIndex(index)** - Returns a reference to the owner at a specified index position in the list of owners that have currently seized capacity units of this resource.
- **Capacity.CurrentOwners.IndexOfItem(owner)** - Returns the one-based index of the first occurrence of a specified owner in the list of owners that have currently seized capacity units of this resource. If the owner has not seized the resource then the value 0 is returned.
- **Capacity.AllocatedTo(owner)** - Returns the current number of capacity units of this resource that are allocated to (have been seized by) a specified owner object.

DateTime Functions

We've added new functions that return a number representing a minute, hour, day, month, year, etc. based on a datetime input. There are also additional functions that convert a datetime to/from a string and those that let you determine which day of the week, month or year corresponds to a given datetime. These functions include:

- **Year, Month, Hour, Minute, Second**
- **DayOfYear, DayOfMonth, DayOfWeek**
- **DaysInMonth**
- **FromString, ToString**

Activity Element Functions

- **ActualDurationFor(entity)** - Returns the actual per batch time duration of this activity for the specified entity. Note that if the activity has not yet been started by the entity, then the duration returned will be based on if the entity was next to start the activity, respecting a sequence-dependent or change-dependent duration type accordingly.
- **BatchesRequiredFor(entity)** - Returns the total number of sequential batches required for the specified entity to perform the activity as per the operation quantity and activity batch size parameters.

Entity Object Functions

- **CurrentOperation** – If the entity object is currently performing an operation, then this function returns a reference to that operation.
- **CurrentActivity** – If the entity object is currently performing an operation activity, then this function returns a reference to that activity.

Workstation Functions

- **ActualSetupTime** – This function may be used as the dynamic selection expression to select the next job at a workstation, or in the selection expression to select a workstation from a list of candidate workstations at a node. It returns the actual setup time duration that would be applied if the associated entity was next to start operation on the workstation, respecting a sequence-dependent or change-dependent duration type accordingly.

Operation Element Functions

- **EstimatedMakespanFor(entity)** - Returns the estimated amount of time for the specified entity to perform this operation, from the start of the first activity to the end of the last activity, assuming the operation's activities will be spanned over (i.e., extended by) any off-shift periods for the operation's primary resource.

New SimBits

We have added three new SimBits to our SimBit library, including:

- **InitializeObjectPropertiesFromATable.spfx** demonstrates how to store and reference object specific data within a table. In this example, information about two servers is stored in different rows in a table. The *Auto-set Table Row Reference* property of an Object Reference column is used to point to the appropriate row.
- **RequestRideFromSameTransporter.spfx** shows how to request the same worker from a 'list' of workers that an entity had utilized earlier in the model for a task later in the simulation. The function Entity.CurrentTransporter is utilized to store information with the entity for later use.
- **UpdateStateInModelFromObject.spfx** demonstrates how a user-defined object can update the state of a model from within the object. The communication between objects in hierarchy is the focus of this example, which is an enhancement of the SimBit 'ProcessModelWithinModel.spfx'.

Performance Slider for Targets (Enterprise Edition)

Within the Targets / Tables / Gantt ribbons in the Enterprise edition, we've added a performance slider to easily allow users to change the upper and lower limits of target performance levels.

Simio Release 4 – Sprint 64 – December 1, 2011

There are two major components to this sprint. First we are releasing the long anticipated **Simio Enterprise Edition** with its Risk-based Planning and Scheduling (RPS) features. This culminates over a year of implementation. While we will continue to enhance the Enterprise feature set (just as we continue to enhance Design Edition), the general release of Enterprise Edition is an important milestone. Second, our "routine" enhancements to the base simulation products continue. Features include resource costing for Workstation objects and added material costing, 'Random' selection rule for selecting between resources, added user customized functions, and more.

Simio Enterprise Edition

Simio Enterprise adds a powerful set of patent-pending features. You can use Enterprise as either a richer, more feature-packed, environment for general modeling, or you can use it to extend your simulation models into daily operational support. Extend the use and lifespan of your previously built Simio models or build and execute new models for Risk-based Planning and Scheduling (RPS). Custom-tailor reports, graphs, and tables for use by schedulers. Reduce your risk and costs by analyzing your schedules in ways never before possible.

Enterprise features include:

<u>Table States</u>
Enterprise users can add State columns to a Data Table, which allows them to update our Data Tables with information. This updating can happen during the model run and then the contents of the Data Table can be exported after the run. This powerful feature provides users with a new method for keeping track of data during a run and also opens possibilities for getting run time data out of Simio into other production systems.
Unlike other Table column types, the table state column values are not inputted by the user at the start of the simulation run. These values are assigned during the simulation run by an Assign step. Table States may also be utilized in an expression. Just as you can define different types of states for an object, such as an Integer State or a String State, you can also define different types of States for a Table column. Other examples include Boolean, List, Object Reference and Level states, to name a few.
<u>Targets</u>
Enterprise users can add Targets to a Data Table to use as another type of output measure. A Target is an expression that will be evaluated at the end of the simulation run to determine if a particular goal is met, such as "Is this target within a user-defined acceptable range?"
A target's Expression property is evaluated at the end of a simulation run and classified according to the upper and lower bounds specified. The user defines these Upper and Lower Bounds and then also specifies what text should be written to the table if the final value falls below the Lower Bound, above the Upper Bound or falls within the Bounds. An example might be to have the column display "On-Time" for orders that fall within Bounds or "Late" for orders that fall above the Upper Bound.
<u>Custom-design a Scheduling Environment</u>
You can define what the scheduling user can see and edit in the Data Tables. The Enterprise user can decide if a scheduling user (using the Scheduling Edition software available soon) or someone accessing the Tables from the Planning tab can view or edit each column of each table. The Enterprise model builder might want to allow other users to see certain columns of a table, not see other columns or even

allow editing of a few columns. This is controlled by two properties on a table column called Visible and Editable found under the Operational Planning property category.

Resource Gantt Chart

Simio Enterprise Edition provides a Resource Gantt chart that graphically illustrates the states of constrained resources over time. It displays each entity that utilizes the resource. Within this Resource Gantt, you can graphically edit the resource's capacity, allowing you to add exceptions to a Work Schedule, such as adding overtime for a resource. The chart also allows the user to see which entities are waiting for the resource.

Entity Gantt Chart

Simio Enterprise Edition provides an Entity Gantt chart that graphically illustrates an entity's usage of resources over time. The user can also see each constraint that impedes the progress of the each entity. This Gantt also displays date-time based milestones and targets for each entity, along with any associated risk measures.

Detailed Logs

Resource Usage Log

This table provides detailed information on each resource and each entity that utilized it. Information such as the start time, end time, duration of use are available for a thorough analysis. This chart can be filtered and sorted.

Resource State Log

This log provides details of the resource states over time. It shows all the resources, their resource state (i.e. Busy, Failed, OffShift), when that resource entered that state and when it left that state. This chart can be filtered and sorted.

Constraint Log

This table shows a record of each constraint that impedes the progress of each entity. It aids in identifying bottlenecks and determining how to best improve a facility's operation. This table will indicate if an entity is being constrained because it is waiting on the capacity of a resource to become available or because it is waiting for an arrival of a resource, such as a Worker or Vehicle. The name of the constraint item is listed, along with a description of that item and when the wait time began and ended. This chart can be filtered and sorted.

Results and Analysis

Target Summary provides a high level summary of Target performance. There are also Risk Plots are available for a graphical analysis of Target performance. After the Risk Analysis is run, the Risk Plots provide confidence intervals and percentiles for the target of each entity. Detailed Results, such as queue lengths and wait times, are available in a pivot grid that can be filter, sorted or exported. This is similar to the Results pivot grid in a Simio model 's Results tab.

There are three reports available to display results of a Plan:

The *Resource Dispatch List* report includes chart and graphical representations for tasks on a particular resource for a given time period. The user inputs the specific Resource for which to display information and the report will display the entity, the start time, the end time and duration that this entity used this resource.

The *Workflow Constraint Analysis* report provides additional information to the Constraint Log within the Entity Workflow window. For each constraint type, the report includes information on the Constraint Item, number of Occurrences, Average Wait and Total Wait times, both in chart and graph formats. Constraints are then also displayed graphically by Facility location.

The *Resource Utilization Summary* report displays all the Resource states of each Resource and the total time it spent in that state. The information can be displayed in both a chart and graphically within the report.

General Simio Features

Random Selection Rule for Seizing Resources

We have added a new rule for selecting from Resource Lists, the 'Random' selection logic. Now, when seizing a resource / worker from a list of objects, you may randomly select between them. This is an enhancement that has been requested from many of our users.

Named Views for Pivot Grids

We have enhanced our Pivot Grid ribbon to include the Add View, Change View and Manage Views options, which users currently have within the View ribbon for saving animation views. However, these various views within the pivot grid allow users to save changes to sorting, filtering, column ordering and column width. Pivot Grids are available within the Results tab for running a simulation, experiment viewing when running multiple simulation replications and within the detailed results area of the Planning tab within Simio Enterprise edition for planning and risk analysis.

Costing for Materials and Workstation Objects

We have enhanced our costing features to include both material costs and workstation costs. The Material element now includes a Cost per Unit that is charged if a quantity of material is consumed. The Consume step includes the option to accrue material costs and a new Material element function, MaterialCostCharged, has been added.

Within the Workstation object of the Standard Library, resource costs have been added that include the Idle Cost Rate, Cost per Use and Setup, Processing and Teardown Cost Rates.

New Costing Functions

Two new functions are available for all intelligent objects that are considered resources, including Capacity.IdleCost and Capacity.UsageCostCharged. These functions will return the Idle Cost and Usage Cost Charged data items for a resource. This information is reported within the results of the simulation run.

User Defined Functions

We have added the ability for users to define their own custom functions within Simio. This can be done within the Definitions tab / Functions panel by selecting the **Function** button. A customized function may include a combination of other functions and/or expressions that the user is interested in viewing or analyzing during the simulation run.

This new capability supplements the previously available Function Tables on the Data tab which are useful for calculating a return value based on interpolating from a set of values. The new Functions allow you to create your own function name that will return the result of any expression you provide. This is particularly useful when you repeatedly use a long expression and you would like to instead reference that expression by a simple name.

Function Tables within the Data tab have been renamed **Lookup Tables** to avoid confusion and the keyword **Function** is no longer needed in front of the Lookup Table name.

Simio Release 4 – Sprint 63 – November 16, 2011

This sprint we continue to enhance the Risk-based Planning and Scheduling (RPS) features of Simio Enterprise Edition. We have also added resource costing to the Standard Library objects and made some UI changes. Our biggest feature in this sprint, however, is the addition of 'table-based' elements through the auto-create features. Now, within a data table, elements and their properties may be automatically created. This opens up many possibilities for importing bill of materials and so much more into the simulation model using tables and auto-create.

Data Binding Enhancements for Table Importing

There are some new enhancements to the data binding features within Simio, including the addition of database binding (i.e., binding a data table to Access, ODBC, Ole DB, Oracle or SQL Server). We've also added Binding Options, so that data within a table can be automatically updated / bound to a table at the start of a simulation run or manually imported.

Auto-Create Elements from Tables

Simio now has the capability of defining Elements and their properties from within Data Tables. This powerful feature enables users to store information, such as bill of materials and many other data items within Excel or other data sources, import into a table and automatically create various Elements.
For any given Element Reference column in a data table, the Reference Type property can be changed from the default value of 'Reference' (which will simply reference an existing element) to 'Create' (which will actually auto-create an element). Each element type has its own properties that can also be populated with a single value, expression or reference to another column in the data table.

Skybox Animation Feature

Within the 3D view of the Facility window, users now have the capability to define a SkyBox, which will place either a static or dynamic background or sky view within the animation. Dynamic options include clouds moving (typically the speed factor should be set to 5-10 or greater to see movement), while static options include various outdoor scenes. Users can create their own SkyBoxes as well. The "library" skyboxes are simply *.zip files with images inside them with the names front, back, left, right, top, bottom. They don't need to have all of those names in them, Simio will use whatever ones it finds. They are located in the Skybox folder under \Public\PublicDocuments\Simio.

SetRow Step Enhancement

The SetRow step within the Processes window has been changed to allow the token to assign a Table Name and Row Number to a specific object or element. This would include object / element reference names as well. This allows the user greater flexibility in assigning table data to specific objects, such as a Server for example.

Costing Enhancements

Resource Cost categories have been added to the Standard Library objects that are utilized as resources. This includes Server, Combiner, Separator, Resource, Vehicle and Worker objects. Resource costs include Idle Cost Rate, Cost Per-Use and Usage Cost Rate. Cost rates are accrued to the entity or owner object for the duration of the idle or busy time.
The Seize step now includes two new costing related properties within the Resource Seizes repeat group. These include the Accrue Usage Costs (True/False) and Per-Use Cost Accrual (At Seize, At Release, Next Station). These allow the user additional flexibility in determining whether a particular seize step will cause costing to be accrued and if so, when the per-use cost will occur.

Simio Enterprise Documentation

While the Simio Enterprise Edition has been available to early adopters for about 10 months, it will be made publicly available very soon. Enterprise may be thought of in two ways:

A new high-end version of Simio containing many valuable modeling features like:

- Specify states that can be added to a table. Table states can be written to during a run, thus updating your tables with output information.
- Add Targets to tables to use as another type of output measure: "Is a target within a user-defined acceptable range?"
- Model "builder" can customize tables for a model "user" controlling what is displayed and what of that can be changed.
- Resource Gantt chart to graphically illustrate states of constrained resources over time
- Entity Gantt chart to graphically illustrate an entity's usage of resources over time
- Detailed Resource Usage Logs that list how long and when a Resource was used by a particular entity.
- Resource State Logs that summarize how long and when a Resource was in each state (for example; Busy, OffShift or Failed).
- Analysis of Targets both deterministic (plan) and stochastic (risk analysis).
- A Constraint Log which aids in identifying bottlenecks and determining how best to improve a facility's operation.

A new tool for Risk-based planning and scheduling

- A very flexible modeling and analysis environment including all of the tools mentioned above.
- Works with Simio Scheduling Edition for deployment of a customized tool for schedulers.

If you are interested in learning more about Simio Enterprise Edition, both an introduction to Risk-based Planning and Scheduling (RPS), as well as the user's guide for the new features are now included in the Simio on-line Help. Look for the major topic named "Scheduling in Simio Enterprise" or look for any of the sub-topics that include the word "Enterprise". Please note that these features are only available in the Enterprise Edition, which requires an upgrade from Team Edition.

Simio Release 4 – Sprint 62 – October 25, 2011

This sprint, like the past several sprints, includes many new features in Risk-based Planning and Scheduling (RPS). We have enhanced our Costing capabilities, have added some features to the Processes window and continue to add SimBit examples to our extensive library.

Processes Window Ease of Use

The Processes window now has auto-scrolling capability, which will make navigating around larger processes much easier. When you want to move a step or a segment connector to another location within a large process, simply move the connector or step to the right or left and the window will auto-scroll. Additionally, the right-click menu for steps has been enhanced to provide options to Disconnect Exit and Disconnect Alternative Exit. These options are available if the step's exit can be disconnected without leaving part of the process disconnected from the Begin step.

Costing Enhancements

We've continued to add costing to the Simio product. With this sprint, Cost per Use and Holding Cost Rate properties have been added to every buffer within the Standard Library objects. These define costs to hold an entity in the buffer. For example, the Source includes these costs for its output buffer, while the Sink includes this costing information for its input buffer. Server, Workstation, Combiner and Separator now have buffer usage / holding costs for all input and output buffers.
Any transporter object, including the Vehicle & Worker objects in the Standard Library, now have a new Transport Costs section that define costs for an entity to use the transporter (cost per rider), as well as a transport cost rate.
The Station element now includes a 'Financials' section with Parent Cost Center, Cost per Use and Holding Cost Rate properties.

New Costing and Population Functions

In addition to costing properties within the Standard Library objects, there have been many new functions added for users to easily access costing information about the model, as well as population information about entities. These include:
For Queue States:
MinimumCostWaiting - Returns the minimum cost that an item accumulated while it waited in the queue.
MaximumCostWaiting - Returns the maximum cost that an item accumulated while it waited in the queue.
AverageCostWaiting - Returns the average cost that an item accumulated while it waited in the queue.
For entities / agents:
Population.CostPerItem.Average – Returns the average cost that was accumulated by destroyed objects of this object's type.
Population.CostPerItem.Minimum – Returns the minimum cost that was accumulated by destroyed objects of this object's type.
Population.CostPerItem.Maximum – Returns the maximum cost that was accumulated by destroyed objects of this object's type.
Population.Index – Returns the one-based index of this object in the population that the object is a member of.
Population.FirstItem – Returns the reference to the first member in the current of this object's type.
Population.LastItem – Returns the reference to the last member in the current of this object's type.
Population.ItemAtIndex(index) – Returns a reference to the member at a specific index position in the current population of this object's type.
Population.IndexofItem(object) – Returns the one-based member index of a specified object in the current population of this object's type. If the population does not contain the object, then the value '0' is returned.

New Costing SimBit

We have added a new Simbit to our extensive SimBit library:

- **Financials.spfx** highlights some of the new financial support recently added to Simio. The model demonstrates modeling Capital Cost, Buffer Holding Costs, Cost on an Entity and how to manually apply costs in process logic. The SimBit documentation contains detailed explanations for how the costs in the Results tab were calculated by Simio.

New Example

We have added **RSPExample.spfx** to the pre-installed examples. While this requires and is primarily intended for use with Simio Enterprise Edition, many of the basic modeling features are available for viewing with other Simio editions.

Simio Release 4 – Sprint 61 – October 3, 2011

This sprint includes many new features in Risk-based Planning and Scheduling (RPS) which is still in the early adopter phase. Since most people do not have access to those features they are not described here. (If you want more information, look to the Simio Products page or contact Simio for a personal demonstration). In addition we have improved code stability (that's marketing speak for fixed some bugs ☺). We have also added features that allow you to run larger models and make better use of your computer resources, as well as additional capabilities for Costing and Resources.

Experiment Replication Runner

Although Simio does not have any built-in memory limits, we were previously limited to the address space permitted by 32 bit operating systems. This limitation was most frustrating when you tried to take full advantage of multiple processors when running a large model. We have made two changes to address these problems.

- Multiple concurrent runs can now be made using separate processes. This will give everyone (regardless of your OS) the full memory available to your OS for use with each replication or scenario being run.
- If you are running a 64 bit OS, Simio can now take advantage of this when running experiments. Since 64 bit has the potential to address a huge memory space, each replication can use the full memory available to your system.

Under **Advanced Options** in **Experiment** you will find new options:

- **Replication Runner** – Provides 3 options to control how multiple replications are run.
    - ***InProcess*** runs replications in the same process as the main Simio executable. This is the fastest mode for small models, but is also the most limited in availability of memory.
    - ***External32*** takes advantage of separate processes for each run, but always uses 32 bit processing, even if you are on a 64 bit OS. This is a good compromise of making more memory available, while preserving execution speed.
    - ***External*** takes advantage of separate processes for each run, using either 32 bit or 64 bit based on your OS. On a 64 bit system, this may run some models slower, but can take full advantage of all available memory. Some very large models may require this setting.

  Note, when you first use either of the external options, you may get a warning message from your firewall protection software which will result in an error in Simio. If you allow Simio access through the firewall (still internal to your computer), it should run normally.
- **Concurrent Replications** – Allows you to limit the number of replications that you will run concurrently on a multiprocessor system. This defaults to unlimited (0), but is appropriate when your model requires more memory than your current hardware and software configuration can provide.

Costing Features

Last sprint we introduced the Cost Centers element for the model, as well as Cost states for each object. This sprint, we have enhanced our costing features to now include Parent Cost Center and Capital Cost to most of the Standard Library objects, as well as Initial Cost and Initial Cost Rate for the ModelEntity. Costing calculations are then done automatically for each static object, such as Source, Sink, Server,

Workstation, Separator, Combiner, and Resource. These costing features are still at the early look phase – feel free to experiment with them and tell us what you think, but don't yet use them in a real project because they are still changing.

Resource Specific Exceptions

It is fairly common that facilities operate with a set of "standard" work schedules, but that specific people or devices might temporarily work a different schedule (e.g. "Joe, can you work 2 hours overtime on Wednesday?"). For many of our Standard Library objects that contain resources, we have added the ability to 'override' a specific schedule's exceptions for that particular resource. For example, let's say that Server1 follows an 'AllServers' schedule, which includes several work day and period exceptions (as defined in the Schedules panel of the Definitions window). Server1 may also specify a particular work day or work period exception that will 'override' those defined with the 'AllServers' schedule.

Simio Release 4 – Sprint 60 – September 12, 2011

Within this sprint of Simio 4, we've spent a lot of time adding new user functions and providing greater flexibility for users for modeling. We think you'll find the Object and Element reference states, in combination with the functions, very powerful in model building. Additionally, we've begun the process of adding Costing mechanisms to the software. This sprint provides you with some costing basics, allowing users to assign costs and specify currency types. Future sprints will expand on the costing basics to provide even greater functionality.

New Functions for Intelligent Objects

We have added many new functions to all intelligent objects (which include all Standard Library objects) as well as some specific functions for particular groups. Below is a list of those new functions:

Current Node - If the entity object's leading edge is currently at a node or the entity is located in a station owned by a node, then this function returns a reference to that node. Otherwise the Nothing keyword is returned.
DestinationNode - Returns a reference to the entity object's current destination node, or the Nothing keyword if the entity does not have a destination.
CurrentLink - If the entity object's leading edge is currently on or at the end of a link, then this function returns a reference to that link. Otherwise the Nothing keyword is returned.
TrailingLink - If the entity object's trailing edge is currently on or at the end of a link, then this function returns a reference to that link. Otherwise the Nothing keyword is returned.
CurrentStation - If the entity object is currently located in a station, then this function returns a reference to that station. Otherwise the Nothing keyword is returned.
BatchParent - If the entity object is currently located in a station, then this function returns a reference to that station. Otherwise the Nothing keyword is returned.
ReservedTransporter - If the entity object has made a reservation to ride on a transporter, then this function returns a reference to that transporter. Otherwise the Nothing keyword is returned.
CurrentTransporter - If the entity object is currently riding on a transporter, then this function returns a reference to that transporter. Otherwise the Nothing keyword is returned.
NetworkDistance.ToNode(node) - Returns the shortest travel distance from the entity object's current location to a specified node using the entity's current network.
NextEntityAheadOnLink - If the entity object's leading edge is on a link, then this function returns a reference to the next entity immediately ahead on the same link. If there is no entity ahead then the Nothing keyword is returned.
NextEntityBehindOnLink - If the entity object's trailing edge is on a link, then this function returns a reference to the next entity immediately behind on the same link. If there is no entity behind then the Nothing keyword is returned.
SeizedResources.NumberItems - Returns the current number of resources seized and owned by this object.
SeizedResources.FirstItem - Returns a reference to the first resource in the list of resources currently seized and owned by this object.
SeizedResources.LastItem - Returns a reference to the last resource in the list of resources currently seized and owned by this object.
SeizedResources.ItemAtIndex(index) - Returns a reference to the resource at a specified index position in the list of resources currently seized and owned by this object.

SeizedResources.CapacityUnitsOwned(resource) - Returns the number of capacity units of a specified resource that are currently seized and owned by this object.
Location.Parent - Returns a reference to the parent object that contains or is associated with the object's current location.
CurrentNetwork - Returns a reference to the entity object's currently assigned network.
Distance(fromNode, toNode) - Returns the shortest travel distance between two nodes using this network.
DirectDistanceTo.Object(object) - Returns the direct (straight-line) distance from this object to another specified object.
DirectDistanceTo.Location (x, y, z) - Returns the direct (straight-line) distance from this object to a specified coordinate location.
Population.Cost.Total – Returns the total cost that was accumulated by destroyed objects of this object's type.
Population.Cost.Average – Returns the average cost that was accumulated by destroyed objects of this object's type.
Population.Cost.Minimum – Returns the minimum cost that was accumulated by destroyed objects of this object's type.
Population.Cost.Maximum – Returns the maximum cost that was accumulated by destroyed objects of this object's type.

Object and Element Referencing

Object Reference and Element Reference state variables have been added to the Definitions tab, States panel of the product. This powerful feature allows users to store the names of elements and / or objects in state variables for use in various properties of objects, as well as functions. For example, the name of a Process can be stored within an 'Process' type of Element Reference state for an entity; then, within the model, the state name can be used when referencing a process. This would enable multiple entity types to reference different processes to be used within the same property field. This feature, along with the various functions that have been added, will allow users much more flexibility in specifying properties and enables 'macro' type modeling.

Costing Basics

The start of some basic costing constructs have been added to Simio, including a new Cost Center element, which can be found in the Elements panel of the Definitions tab. Cost Centers can be hierarchical by specifying parent cost center information. The Advanced Options under the Run ribbon now also includes several Financials options, including default currency and exchange rate values. Objects now include a 'Cost' level state variable, which permits the cost or cost rate to be changed by using the Assign step in the Processes window. There are multiple costing functions available as well.

Additional costing functionality, including automatic costing calculations within Standard Library objects, will be implemented in future sprints.

New SimBits

We have added two new SimBits to our SimBit library that demonstrate the object referencing states.

- **Object ReferenceOnEntity.spfx** demonstrates how one of the new functions, Location.Parent, can be used to store where an entity is currently located. This information is then later used in decision making logic.
- **SeizingSameResourceFromList.spfx** shows the use of an object reference state to store the name of the last seized resource that an entity was allocated (from a list in this case). The object reference state is then used in the next Server to seize the same resource, using the state name.

Importing tables into repeat groups

Table names and schema may now be converted from table format to repeat group format by using the 'Convert to Repeat Group' button in the Table ribbon. By selecting this, a table of the same name is created, with repeatable properties for each of the columns. The name of the table is then changed to be 'TableName_Original'. This feature allows the mapping of table data from a higher level in object hierarchy.

Persisting user settings

Users often customize the appearance of components in your model like column widths, table tab ordering, filtering on a particular column and setup of the categories within the Processes window. When a model is saved, your user-specified settings are now saved with the model so that when you open the model again, your changes are restored.

Load recent project

A 'Load most recent project at startup' button has been added to the Save window. If this box is checked, your project will automatically open when starting Simio.

Compatibility Issues

Objects with base classes who have newly introduced items with the same identifiers as items already in the object will cause less problems than they used to. Refer to the Known Issues.pdf file for more details.

Simio Release 4 – Sprint 59 – August 18, 2011

This past sprint includes several features that users have requested. We have also spent a good deal of time on object and element referencing, which should be available in the next sprint.

List Creation using Right Click in Facility Window

You can easily create your node, object, and transporter lists by simply right-clicking on the object(s) within the Facility window and adding them to (or deleting them from) an existing or new list. In previous Simio versions, a List was created by going to the Definitions window, Lists panel, specifying the list type and then entering all of the object names for that list. Now, you can individually or group select the items you'd like in a list and add them to an existing list or create a new list. Based on the type of objects selected, Simio determines the type of list that can be generated.

Create step and Separator object include Copy to a new Object Name

Prior to this sprint, when a copy of an entity was made using either the Create step or the Make Copies option of the Separator object, the copy was of the same type as the original entity. The option is now available to change the type of entity that is created. The information from the original entity is copied to the newly created entity, including state values, assigned network and destination, as well as table references. The size values, entity picture and any properties will be assigned from the new entity type.

Simio Release 4 – Sprint 58 – July 29, 2011

Our past several sprints have been spent on many enhancements to the Simio software, as well as additions to our new Enterprise edition. Enhancements such as being able to see the Facility view of an object in hierarchy will make model building and model verification easier. We've made significant changes to our schedules editor to make it easier to use and have added a SimBit search mechanism.

Facility View Visible within External View in Hierarchy

You can now "see" the behavior inside your objects. We have spent several sprints enhancing our hierarchy capability so that what is shown in the Facility window of a model can be 'externally visible' in the External view of a model object. This allows the user greater flexibility and visibility of the lower level objects within a hierarchical object.

For example, if you have an object that consists of three servers and multiple paths between these servers, you will now be able to make part or all of the objects visible within hierarchy. When you place this object into a model, the servers and paths now appear and you can see the entity flow within the new object. Binding of external input and output nodes has also been made easier by using the right-click menu of an input or output node within the Facility window.

Tally Capability in Basic and Transfer Nodes

Both Basic and Transfer nodes now have the capability to have multiple tallies calculated when entering or exiting the node. This provides the ability for many of the standard library objects, such as Source, Server, Combiner, Separator, Sink, etc. to have this capability as well, as they all have basic and/or transfer nodes.

Improved Excel Import

We have improved our capability for importing data from Excel. Users are no longer required to have Excel installed on their computer to utilize the import capabilities within Simio.

Sample SimBit Solutions

A new Sample SimBit Solutions icon has been added to the Navigation window, which will open a SimBits search window. SimBits can now be searched by category and keyword to more easily find examples of particular modeling concepts, standard library objects or process steps. Basic Search provides the user flexibility to type in their own keywords or steps to search, while Advanced Search utilizes categories and up to 3 keywords from an extensive pre-defined list to search for various SimBit examples.

Repeating Arrivals Tables

We've added additional capability for arrival tables within the Source object, such that multiple events per time slot may be modeled. This provides for greater flexibility for groups of arrivals and modeling no-shows and arrival time deviations.

Turnaround Method

We have added capability within each of the entity type objects, including ModelEntity, Worker and Vehicle, to specify a network turnaround method to be used at nodes on bidirectional links. The options for Exit & Re-enter, Rotate in Place and Reverse are now available when an entity changes direction on a bidirectional link. This provides the capability for modeling various types of vehicle / worker movement at a node.

New Schedules Interface

Our previous schedules interface has been changed to be more consistent with our tables interface. Multiple 'day patterns' can be defined with start time, end time, duration and capacity. Day patterns are then specified within a weekly work schedule to define a schedule that can be used for Workers, Resources and many more of the Standard Library objects. Work day exceptions are available for modeling overtime or single holidays. Work period exceptions can be used for modeling planned maintenance or vacations.

Token Wait Action property added to Transfer, Park, and UnPark steps

For more flexibility in modeling various situations, we have added a Token Wait Action property to a number of our process steps, including Transfer, Park and UnPark. The options for the token are WaitUntilTransferring or WaitUntilTransferred. These options will allow users more control over token release timing. For example you could use this new capability to make a token wait at a **Ride** step until the entity had been loaded, instead of the token always just immediately exiting the **Ride** step after initiating the ride request.

Population Functions

People often want to look at function describing an entire population (e.g., an instance) rather than only a specific member of the population (e.g., a dynamic object). For example you might want to look at the instantaneous or average number in system, time in system, capacity, or utilization for an entire group of entities, workers or vehicles rather than the individual entity. New Population functions have been added in the expression editor to make this more accessible.

Enhanced Trace

To increase user debugging capabilities, we've added additional detailed trace for Ride, Pickup, PlanVisit, SelectVisit and SelectDropoff steps when evaluating a candidate request for selection. If a candidate is rejected for some reason, detailed reasoning behind the rejection is displayed within the Trace window. This will allow users to more easily determine why a particular entity was not picked up by a specific worker and/or vehicle. We have also added more Timer-related trace.

Execute Step Behavior

The Execute step behavior has changed to be more intuitive to what customers expect. The *Token Wait Action* of 'None (Continue)' schedules the specified process on the simulation calendar as an early current event and the original token immediately exits the step. If a token passes through multiple Execute steps in sequence, once the original token stops, the execution of the specified processes will occur in the same sequence that they were specified (FIFO order). In previous software versions, processes were executed in reverse (LIFO) order. The *Token Wait Action* of 'WaitUntilProcessCompleted' will immediately execute the specified process. If the process is completed with no delays, the original token immediately exits the Execute step and continues processing. Otherwise, if the specified process

did have delays, then the original token is held at the Execute step until the process is completed. In previous software versions, even if the executed process had no delays and was immediately completed, the original token's departure from the Execute step was scheduled on the event calendar rather than continuing its processing 'in-line'.

New SimBits

We have added four new SimBits to our extensive SimBit library that demonstrate some of our new capabilities.

- **RotatingVehicleExample.spfx** shows the use of the Vehicle from the Standard Library as a robot that rotates and moves entities between various locations.
- **ServerWithTransferInConstraints.spfx** demonstrates the Transfer-In Constraints that has been added to processing type objects, i.e., Server. An expression is evaluated before an entity transfers in to a node to be sure the entity can enter a node, otherwise it continues on the link to another location.
- **TurnaroundMethods.spfx** shows difference between the three different Network Turnaround Methods that can be specified for entities. These include Exit & Re-enter, Rotate in Place and Reverse.
- **UserDefinedListState.spfx** shows an example of the user defined list states for generating time persistent statistics on the state of multiple servers.

Additional Symbols

We have enhanced our symbol library with new symbols, including a robot, forktruck and various pieces of medical and office equipment. Additional textures are also available.

Simio Release 3 – Sprint 54 – April 18, 2011

In this sprint, new features make adding logic to your model easier. Many of you asked for processing time based failures and more flexible arrival tables. There is now more capability to display and write out information related to the simulation run, as well as additional standard processes that can be accessed for writing customized statistics. And of course, we are continuing our development work on Enterprise edition, our Model-Based Decision Support product to be introduced in the coming months.

Processing Time Based Failures

We have enhanced the Server, Combiner, Separator, and Workstation objects in the Standard Library to now include a 'Processing Time Based' failure type. If that type of failure is defined, then the Uptime Between Failures is considered to be processing time only instead of total elapsed time.

Enhanced Arrival Tables

We've added two new properties to both the Source object and the Timer element to allow for deviation around the arrival time for table-based arrivals. A deviation property, as well as a 'no-show' percentage will provide the user much more flexibility in defining arrival tables. This expands the use of arrival tables to stochastic applications in addition to deterministic arrivals.

Run Functions

We've added a number of simulation 'run' related functions including:

Run.EndingTime - Returns the time in hours that the simulation run is scheduled to end. If the run's ending time is defaulted to 'Infinite', then this function returns the largest possible floating-point value.

Run.ExperimentName - Returns the name of the experiment if running in experiment mode. Returns an empty string if running in interactive mode.

Run.Mode - Returns the mode that the user is running in, including Interactive = 1, Experiment = 2 and Operational Planning = 3 (for Enterprise edition only).

Run.ModelName - Returns the name of the main running model.

Run.ProjectName - Returns the name of the project containing the main running model.

Run.RandomnessDisabled - Returns whether random sampling in the simulation run is disabled.

Run.ReplicationNumber - Returns the scenario replication number that the run corresponds to if running in experiment mode. Returns 0 if running in interactive mode.

Run.ReplicationsRequired - Returns the scenario replications required if running in experiment mode. Returns 0 if running in interactive mode.

Run.ScenarioName - Returns the name of the experiment scenario if running in experiment mode. Returns an empty string if running in interactive mode.

Run.TimeNow - Returns the current simulation time in hours.

Run.WarmUpPeriod - Returns the time period in hours after the beginning of the run at which statistics are to be cleared. Applies to runs in experiment mode only.

Conveyor Speedups

Non-accumulating conveyors will now run significantly faster when congested.

OnRunEnding, OnRunWarmUpEnding Standard Processes

We've added two new standard processes that are can now be called during the simulation run by the user. These are:

OnRunEnding – If running in interactive mode, then this is executed when the Stop or Reset button action occurs. If in experiment mode, then when the ending time has been reached or calendar events have been exhausted such that the model run is now ending and final statistics are about to be reported.

OnRunWarmUpEnding – If running in experiment mode, if a warm-up period has been specified and is ending, this is executed right before all statistics are cleared. Typical uses of this process may be to do some custom warm-up statistics clearing. For example, you may want to clear a state variable's value that is being used as a custom statistic or take a snapshot of some statistic values at the end of the warm-up period before Simio clears them.

Run Ending Add-On Process

We've added a new Run Ending Add-On Process Trigger to each of the Standard Library Objects to allow users to add process steps, such as a Write step, when the simulation run is about to end.

Warning Notifications on Standard Library Objects

We've added warnings to several of our standard library objects, including:

BasicNode and TransferNode - Now displays warning if auto-destroying entity because failed to transfer onto an outbound link.

Workstation, Vehicle, Worker - These objects are singular resources that are designed to only have a resource capacity of 0 or 1 (Workstation & Worker) or a fixed capacity of 1 (Vehicle). If logic attempts to assign an invalid resource capacity to one of these objects, a warning is now displayed before the capacity is automatically adjusted to a valid value.

New SimBits

We have added two new SimBits to our SimBit library:

- **MoveASeizedObject.spfx** shows the use of the new Move step and how it can be used to seize a worker at a given Server, then move the worker to multiple other locations for delay activities.
- **ScheduledMaterialArrivalsWithDeviation.spfx** shows the new feature on the Source object (also available on Timer element) that allows for deviation around table arrivals.

New Capability for Object Builders

Within the Definitions window, Properties panel, the switch value of a property can now be based on multiple values. For example, a property may become visible to the user when the value of another property is set to 1, 2 or 5.

Simio Release 3 – Sprint 53 – March 28, 2011

In this sprint we think you'll like the new user driven enhancements that we've added to the software. Most of the items below were requests from current users and we're excited to be able to respond with this sprint!

Move Step

We've added a new Step for use in the Processes window. The Move step can be used to move a resource object that has been seized from one location to another. This step is very powerful in many industries where you have workers that are required for tasks at multiple locations.

Warnings and Notify Step

We have added a warning system that allows the user to define various levels of warnings to show during a simulation run. Warnings can be written to the trace window or displayed in a pop-up box. The warning level can be selected within the Advanced Options button on the Run ribbon.

Additionally, users may generate their own warnings and messages using the new Notify step for use within the Processes window. The Notify step allows users to write custom messages and warnings in a specific format to be displayed to the end user when something of interest occurs in the model.

Finally, we've added a few internal warnings that will help alert you to a potential problem. These include when the threshold number of objects has been exceeded, as well as when two objects have entered a node from two separate bidirectional links and a potential deadlocking situation has occurred.

New SimBits

We have added two new SimBits to our extensive SimBit library:

- **SearchTableUponEnteringObject.spfx** shows the use of tables and the Search step to provide information about a server once the entity is within the server. This SimBit is also modified and used with the HierarchyWithTables SimBit.
- **HierarchyWithTables.spfx** shows the use of tables within a submodel.

Miscellaneous Cool Things

If you turn off the visibility of a feature, such as node, queue, or label, in the View ribbon, it will no longer be editable. Items must be visible to be able to be selected, moved or deleted.

We've added a thin red notification strip in the results and experiments views that indicates when information has gone 'out of date' due to changes in the model's logic or data.

If you are running an experiment and have a problem with one of the replications, you may want to be able to run that particular replication to find the problem. Now, you may specify a replication number in the Facility window under Advanced Options button / Randomness (on the Run ribbon) and view that particular replication interactively.

Simio Release 3 – Sprint 52 – March 4, 2011

In this sprint we've improved execution performance as well as spent some time enhancing our on-line help, SimBits and examples. We are also continuing our efforts into new features for Model-Based Decision Support that will be announce in the coming months.

Performance Improvements

We have made changes to the Simio engine to increase runtime performance. You should notice that links (especially conveyors) now run noticeably faster.

Output Analysis Enhancement

We have changed the format for Scenario / Result reporting in output analysis tables to be easier to read and work with. The prior linear format with all of the Responses shown in a single column for each scenario has been replaced by a matrix format with the Responses shown as separate columns for each scenario.

Disable Randomness Function

We've added a new **DisableRandomness** function that can be used in expressions and logic to indicate whether randomness in the current simulation run is disabled. This can be used to change your logic if randomness is used or not. For example, you may want to have a process time average of 10 minutes when randomness is on, but use a more conservative value of 11 minutes when randomness is off. Using the DisableRandomness button may be useful in debugging as well.

Additionally, if the 'Disable Randomness' option is being used, for any standard library object with 'Reliability Logic' support, the reliability logic will be automatically disabled and no failure occurrences generated.

New SimBits

We have added 2 new SimBits to our SimBit library. They include:

- **ExamplesOfConveyors.spfx** demonstrates the use and effect of the different conveyor properties to define various types of conveyor systems.
- **SelectSpecificVehicle.spfx** shows how to request different vehicles for different part types in the system and how to use the Selection Condition property.

New Example

We have added a new Example to our library of complete models that demonstrate how Simio works. It includes:

- **HospitalEmergencyDepartment.spfx** demonstrates an emergency department operation and includes registration, triage, radiology, billing and 6 rooms and beds.

New examples are now accompanied by pdf descriptions and help topics similar to those that are included with SimBits.

Simio Release 3 – Sprint 51 – February 11, 2011

This release focuses on customer evaluation and learning. We've listened to your requests and have incorporated some new features and a new product edition. To help your colleagues evaluate Simio, we've added an Evaluation mode with modeling limits. To help current users, we've added new SimBits, an example and additional flexibility in a few areas.

Fast Forward Enhancements
The Fast Forward operation has been enhanced to run up to 6 times faster than in previous versions.

Conveyor Enhancements
The Conveyor object within the Standard Library has been enhanced to include additional flexibility in defining the conveyor cells. The conveyor spacing may now be defined in terms of a fixed number of cells or a fixed cell size. Auto alignment options for the first entity entering the conveyor are also available.

Auto-Scrolling
Navigation within the Facility window has been enhanced by auto-scrolling feature. If you are dragging an object and attempt to move past the edge of the 2D/3D view, the view will automatically scroll in that direction.

Simio Evaluation Edition
Simio Evaluation Mode has been enhanced to better support its use for both evaluation and training. It retains the ability to build models of any size and complexity (without saving) so all product features may be explored. But now we have added the ability to save small models or model fragments so that you can share your work with others. This new functionality makes it useful for training classes as well. Although the evaluation version may be used within universities, it does not support building all models featured in the Simio books so we still recommend using a student license to permit more flexibility and student project work.

Simio Evaluation includes the same user interface/ features as the Academic versions but has the following limitations:

- Object count limit of 18
- Process steps count limit of 15
- Models in a project limit of 3
- No user-customized (.NET) extensions
- Experiments are available only if the model is within the above limits

If the above limits are exceeded, users can still build and run large models, however there is no ability to save those larger models.

New SimBits
We have added 3 new SimBits to our extensive SimBit library. Two of the SimBits demonstrate bidirectional link options, while the other demonstrates object building and the process industry. These include:

- **PathSelectionRule.spfx** shows two identical systems using a bidirectional link and demonstrates the difference in entity flow when the Traffic Direction Rule is changed.

- **BidirectionalLinks.spfx** demonstrates the use of unidirectional ‘bypass’ links to minimize deadlocking on bidirectional links.
- **MassFlowProcessing.spfx** uses custom objects and flow rates to move material from one stockpile to another.

New Example

We have added a new Example to our library of complete models that demonstrate how Simio works. It includes:

- **ClaudesPizza.spfx** uses custom objects integrated with the Standard Library objects to demonstrate the operations within a pizza restaurant.

Simio Release 3 – Sprint 49 – January 5, 2011

Happy New Years to all of our customers!! We hope you had a wonderful holiday. We start this new year off with a new software release which includes several new options for giving the user additional flexibility within vehicles and workers, as well as processes.

Parking Options for Vehicles and Workers
Two new options have been added to both the vehicle and worker objects for parking. Parking a vehicle or worker is useful to avoid blocking other travelers on a network. It also allows for better animation of workers and vehicles while they are being utilized at a node. The options include:
Park While Busy – True / False
Park For Load / Unload – True / False

Trace Enhancements
The trace for several steps, including the PlanVisit and SelectVisit steps, has been enhanced for better user debugging. Newly added trace includes the details on 'why' if a step's visit selection failed.

Status Plot Enhancements
Status plots have been enhanced to include keeping many more points on the plot than previously before (we used to keep approximately 100, we now keep upwards of 15000). The speed at which status plots update has also been increased.

Process Element Additions
Within each process, there is a property window that includes information about the process. We have added a new *Token Actions* property category which includes the following options:
OnAssociatedObjectDestroyed – End Process / Continue Process
OnAssociatedObjectTransferRequested – End Process / Continue Process
A process also now has a *NumberTokensInProcess* function. This can be referenced by using ProcessName.NumberTokensInProcess.

SPF File Support
Simio stopped generating and directly reading the SPF project file format quite a while ago. Until now, Simio contained a conversion utility to convert the old format to the newer, more compact and flexible SPFX format. That conversion capability has now been moved from the Simio.exe into a separate utility. If you need to convert SPF files you can use Simio software version 3.48 or earlier, or download the stand-alone conversion utility from:
http://www.simio.com/downloads/public/software/ConvertSimioSPFFileToSPFX.zip

Simio Release 3 – Sprint 48 – December 3, 2010

This release introduces a combination of both "fun" and "practical" features. On the "fun" side we have provided some powerful new camera features that will make your 3D animations more compelling to watch and fun to demonstrate. We have also continued our efforts to make the Standard Library more powerful by adding support for using secondary resources directly within our Standard Library objects without using Processes. Responding to frequent requests, we have also added more flexibility in animating queues, as well as the ability to add and delete vertices to existing queues, links, poly lines, and curves.

Camera Position Enhancements
We have added significant new controls for manipulating the 3D camera position and orientation. New controls include:

- Ctrl+Scroll wheel changes the viewing angle of the camera
- Shift+Scroll wheel moves the camera straight up and down
- Up arrow moves forward
- Down arrow moves left
- Left arrow rotates left (or moves left in top-down mode)
- Right arrow rotates right (or moves right in top-down mode)
- Shift+Left arrow moves left
- Shift+Right arrow moves right
- [W] key sets the camera 1.7 meters off the ground, with the angle pointing almost straight out (i.e. "walking mode")

Camera Tracking Enhancements
The camera can now be attached to an object in the system and move with the object, track the object from a fixed position or follow behind/sit in front of it.

- A new **Track With Camera** option has been added when you right-click on a dynamic object while running.
- A **Camera Placement** button has been added to the View Ribbon under Camera Tracking. This allows you to provides four choices on how you would like the camera to follow (or track) the specified object.
- A **Change Item** button has been added to the View Ribbon under Camera Tracking to allow easy change between potentially tracked objects.

Secondary Resources
Within the Server, Combiner and Separator objects, we've added secondary resources that can be specified within the Facility window, instead of using the Processes window and Add-On Process Triggers to seize and/or release resources and workers.

- Under the category Secondary Resources, for simplicity you can specify a single resource (or list member) that is used for processing at the object.
- When more flexibility is required, you can specify multiple resources that can be seized and released at various times within the object (upon entering, before processing and after processing).

Queue Enhancements

- Animated queues may now be specified as **Point Queue** or **Oriented Point Queue**, in addition to the original line queues.
- In an Oriented Point Queue you can change the entity orientation (the direction they face) for each point on the queue.
- Users may also now add vertices to the middle or end of an existing queue.

Vertices Enhancements

Users can now add vertices to an existing polyline, curve or link.

- By simply highlighting a line or curve, a vertex can be added to the end or middle of that line or curve.
- With links, simply highlight the link and add a vertex to be placed anywhere along the existing link.

Sketchup Version 8

We've added support for the latest Sketchup version. Now you can both directly import local files as well as download Google Warehouse files that have been saved with the latest Sketchup.

Random Number Stream Support

Simio has always had an unlimited number of random number streams and the ability to specify the stream number as an optional parameter on distributions (like **Random.Exponential(mean [, Stream])** – we have now made this more clear in the Expression Editor pop-up help. In addition we have now added a Random Number Stream property to the Decide step and the Node object (routing logic) so you can override the default stream number (0) when necessary.

Math.NaN and Math.IfNaN(value)

We've added the math functions NaN and IfNaN(value) that will return the constant NaN (Not a Number) and will evaluate whether a value is NaN. This can be useful with the Math.If() function when there isn't a value that is returned from the If.

Tables and Candidate Keyword

You can now address tables that are bound to candidate objects in expressions using the Candidate keyword. For example, "Candidate.TableName.ColumnName > 100" could be used within a Selection Condition for selecting a resource from a list if the resources are bound to given rows in the table (as opposed to using the Candidate keyword to apply to the executing entity).

Simio Release 3 – Sprint 46 – October 30, 2010

This interim release has only a few new features because we are now putting a significant effort into new features for Model-Based Decision Support. You will see that effort paying off in the near future. But in the interim, you should find the following new features to be valuable.

Read/Write handles string states and string expressions

The Read Step and the Write Step now support reading and writing to and from String states and DateTime states in Simio, as opposed to just numeric States.

Disable Randomness

A Disable Randomness button has been added to the Run ribbon. Disabling randomness is often useful to facilitate verification and debugging, and for generating deterministic schedules. If randomness is disabled, any random distribution reference will return the expected value instead of a random variate. Additionally any probabilistic selection in the logic (e.g., probability based Decide step, link selection weights, the table property RandomRow function, Random selection goal in Route step) will always select the choice that includes the 0.5 (or "50%") cumulative probability.

Name Expression Keyword

We've added the function 'Name' for objects which will return the name of the object in string format (i.e., "Server1"). This can be useful for writing out useful information to floor labels and/or files.

View Filters on Data Table

Filtering for the data grid in tables has been added. Now you'll see a small filter icon on each column, and a bar at the bottom to turn the filter on/off, see your most recent filters, and edit the filter. This makes it easier to view important subsets of a large data table.

Math.IF

We have added the capability to test multiple conditions and return one of a defined set of expressions. The 'Math.If' function has been expanded to now accept multiple test, value pairs using the following format: Math.If(test1, value1, test2, value2, test3, value3, …, otherwise value). For example, Math.If(Is.Airplane, Airplane.OrderID, Is.Order, Order.Id, Is.Part, Part.OrderId, 0).

New SimBits

We have added 4 new SimBits to our extensive SimBit library. Three of those SimBits illustrate the use of tables and the repeating group property interface, while the other demonstrates the use of the expanded Math.If and String States. These include:

- **TableReferenceInRepeatingProperty.spfx** uses a table to store data about the entities and assignments that will be made, as well as the worker type to seize for transport.

- **RelationalTablesInRepeatingProperty.spfx** shows multiple workers being used at various workstations based on the entity types that are processing. The worker and entity data is stored within relational tables.
- **SourceWithCustomEntityTypeProperty.spfx** uses a 'sub-classed' Source to allow user-defined entities / probabilities to be specified for entity creation.
- **StringStates.spfx** uses the string states and the new Math.If capability to graphically show the status of entities being processed.

Data Validation

We have taken advantage of the new data types (like Integer and Boolean) to enhance our data validation. For example if you try to assign a value of 2.56 to a CurrentCapacity, you will now get an error because CurrentCapacity is defined as an Integer. See the Known Anomalies on the Simio Start page for important conversion notes.

Simio Release 3 – Sprint 45 – October 10, 2010

Responding to several user requests, we have added comprehensive support for String States as well as several new numeric state types. We have also enhanced Vehicle behavior to address a common modeling need. And we have added new capability to Repeat groups that will be particularly useful for people building custom hierarchical objects.

States Ribbon

We've added several categories of states to the Discrete group, including **Integer**, **Boolean**, **DateTime** and **String**. The Real, Integer, Boolean and DateTime states can be scalar (default), vector or matrix type states. The String state is scalar. All discrete-change states can be changed by assignment logic at discrete times during a model run.

String Functions

Based on the new String States that have been added, there are also a number of string functions that may be used, including:

String.Length(string) – returns the length of a string
String.Contains(string, substring) – return true if substring is in string, otherwise false
String.Substring(string, startindex, length) – returns a substring of string starting at startindex and of the specified length
String.FromReal(value) – returns a string representation of a real value (String.FromReal(6) turns into "6")
String.ToReal(string) – returns a real value representation of a string (String.ToReal("6.2") turns into 6.2)
String.FromDateTime(value) – returns a string representation of a datetime value based on the start date and time of the simulation (String.FromDateTime(6) turns into "October 1, 2010 06:00:00" if the start date of the simulation is "October 1, 2010 00:00:00")
String.ToDateTime(string) – returns a datetime representation, in hours, of a string based on the start date and time of the simulation model (String.ToDateTime("October 11, 2010") turns into 0 if that is the start date of the simulation, 24 if the start date is October 10, -24 if the start date is October 12, etc.)
String.Format(string, arg1, arg2, …) - returns a new string formatted with the specified format string and arguments. We follow the .NET string formatting syntax here: String.Format("Hello {0}, you are {1} years old", "Friend", 30) turns into "Hello Friend, you are 30 years old"
String.Compare(string1, string2) – returns 0 if the strings are equal, -1 if string1 < string2, 1 if string1 > string2
String.CompareIgnoreCase(string1, string2) – same as above, but ignores casing in comparison, so String.Compare("fred", "FRED") does not return 0, but String.CompareIgnoreCase("fred", "FRED") does return 0.
Strings in Expressions – Simio supports '+' for strings such that "Hello" + " World" turns into "Hello World". We also support the logical comparison operators, such as '==' or '>'. You can now use 'MyStringstate=="Red"' or 'MyStringState > "A"' in an expression

Vehicle and Worker Behavior

We've changed the Vehicle and Worker object behavior for accepting entity reservations based on many customer needs. A Vehicle or Worker with *Ride Capacity* > 1 will now accept multiple reservation requests if all of those requests are originating from the same pickup location. For example, suppose a Vehicle with Ride Capacity = 3 accepts a request to pick up a rider at Node A. While traveling to Node A, that Vehicle will also accept up to 2 more pickup reservation requests (up to its ride capacity) if those requests are also originating from Node A where the vehicle is already headed to do a pickup. Previously, the Vehicle would not accept any more pickup reservation requests after the first one.

Property Repeat Groups

We've enhanced our property repeat group capability by adding the ability to reference repeat groups and their properties similar to referencing tables. This feature should be very beneficial in object building.

For example, let's say you have a repeat group named MyRepeatGroup with two properties within the repeat group, MyTime and MyEntityReference. You may now utilize the SetTable step with the Table Name set as 'MyRepeatGroup'. You may also utilize other table functions, as shown below, in expressions.

- Use "MyRepeatGroup.MyTime"
- Use "MyRepeatGroup[x].MyTime"
- Use "MyRepeatGroup.MyTime.RandomRow"
- Use "MyRepeatGroup.AvailableRowCount"
- Use "MyRepeatGroup.MyEntityReference", as a reference property in places that take an entity reference

You can now specify a table name in a repeat group property. If a repeat group references a table, then when asking for the number of rows in the repeat group (either for traversing the repeat group in a step, or asking for it in an expression, such as "MyRepeatGroup.AvailableRowCount"), the number of rows of the table will be returned. Additionally, when the engine traverses each tuple in the repeat group, it will implicitly map to a set table reference to the corresponding row in the table. Therefore, any table reference properties in the single repeat group referencing tuple, such as "MyTable.Property", will resolve correctly.

Simio Release 3 – Sprint 44 – September 20, 2010

This sprint was mainly focused on improving the execution speed - we have made significant improvements in many areas. A few other software enhancements have also been made.

Model Execution Speed
Execution speed is significantly faster. Most users will see their models execute from two times faster (e.g. runs in half the time) to ten times faster or more. The enhancements were implemented throughout the product so your performance improvement will vary depending on exactly what you are doing. One notable enhancement is that experiments with many replications may now run 5 to 10 times faster. Also models with many concurrent entities will be much faster. We even had one customer model that runs over 70 times faster (but that level of improvement will probably be uncommon).

Table Column Handling
Two features have been added to make it easier to work with tables. First, buttons have been added to the Table ribbon to **Move Right** and **Move Left**. These can be use to highlight a column and shift it to the new desired position. A second feature was added to allow you to change the type of a column that has already been added. Look for the **Change Type** button on the Table ribbon.

Symbol Sizes
Have you ever downloaded multiple 3D symbols into your model and only after you saved you found that your model size increased by tens of megabytes? Although you can use advanced features of the Google 3D Warehouse search to limit the size of objects you search for, other than that there is not much help available. But at least now you can see the symbol size at the bottom of Simio's import dialog. And you can look at all the symbols in your project and determine the size of each one to perhaps replace a particularly large one. Use the Navigation window to select your project (named MySimioProject by default), then select the Symbols panel. This will display all your symbols and their sizes as well as provide options to add and delete symbols.

Interrupt Step
We added a new Interrupted Process Action to provide an additional choice for what the interrupted token should do next. The new **EndDelay** option specifies that the token will end its delay and exit the Delay step once the new token created to handle the interruption ends its processing.

States Ribbon
We made some cosmetic changes to the States ribbon to clarify what you are getting. The states are now divided into two categories, those which change discretely (e.g. only by assignment) and those that may also change continuously. The **Discrete** group includes the newly renamed **Real** state and the existing **List** state. The **Continuous** group includes the existing **Level** state and the newly renamed **Level with Acceleration** state.

Error Detection and Reporting
Over the last few sprints we have improved Simio's error detection. We understand that it is painful to get a new error when you load a "working" model into newer software. But in general, this means that your model was not really working as you expected – it likely had a subtle bug. But we have also improved the content and clarity of many errors so they will now be easier to find and fix.

Simio Release 3 – Sprint 43 – August 21, 2010

This sprint was mainly focused on improving the execution speed. We are not yet done, so we have not done any detailed analysis yet, but you may already notice improved speed in many different areas. In addition the following changes were made:

Model Initialization
Improved model initialization especially regarding the use of Monitor and StateStatistic elements and states of dynamic objects.

Table Binding
We enhanced binding in two ways. First, we now show information about what data source the table is bound to. Second, if the bound file isn't found (like if you open the project on a computer with a different directory structure), then we will now also look next to the project file for a file of the same name. This should make it easier to deploy solutions with bound data.

Error Messages
The text and format of many error messages have been improved to clarify and help resolve the problem.

New SimBit
RoutingWithoutPaths illustrates how to move entities directly from node to node without traveling on a link.

Simio Release 3 – Sprint 42 – August 8, 2010

This sprint culminates almost eight months of progress. The next 25 pages describe the hundreds of new features that have been added since Version 2. In this latest sprint we have a few important features as well as bringing together the last 11 sprints into a cohesive major release. Overall, you will find that Simio Version 3:

- empowers novice users through additions and enhancements to the standard library,
- brings greater flexibility to power users with new process steps,
- is easier to learn with improvements to UI, SimBits and learning materials,
- has improved experimentation with ranking and selection tools, and OptQuest,
- provides better understanding of output data with SMORE plots, resource statistics, units support and better reports, and
- gives advanced users sophisticated tools for search, remove, and interrupt, many new functions, and an improved API.

In Sprint 42:

Undo/Redo

It was a long time coming and it took many sprints of effort, but you prioritized it so we have it – comprehensive undo and redo capability across the product to make it easier to safely explore and to recover from any accidents. You will notice undo and redo buttons on the quick access toolbar on the upper left. Ctrl-Z and Ctrl-Y also work.

OptQuest

Simio now includes OptQuest from OptTek Systems. OptQuest is the marketing leading tool designed to use simulation to search for optimal solutions. OptQuest uses methods that integrate state-of-the-art metaheuristic procedures, including Tabu Search, Neural Networks, Scatter Search, and Linear/Integer Programming, into a single composite method. In short, Simio is great at modeling your problem and evaluating a proposed solution. OptQuest is great as using that same model to evaluate lots of possible solutions to help you make the best choice.

OptQuest is included in demonstration mode for all users. Commercial users can obtain OptQuest activation for a separate fee. Student and Academic licenses include the full unlimited-size version of OptQuest (limited to non-commercial use).

Report Categories

We revised our reporting categories for clarity. The new categories include: Content, Throughput, Capacity, FlowTime, and HoldingTime. Taking advantage of the pivot table capabilities to sort and filter on these categories will help you more effectively analyze your data.

Experimentation

Limited support is now provided for including non-numeric Controls (e.g. Model Properties) in an experiment.

Queue Names

Each station element has queue holding the entity objects physically located in the station location. The name of that queue used to be 'InProcess'. However, as we enhanced Simio's resource state tracking and utilization statistics and the term "Processing" became more prevalent, confusion related to using 'InProcess' for that queue name increased as well. The old queue names still exist and are valid, but are no longer in the interface or documented. Instead you will now see:

- **'InProcess' queue of a station was renamed to 'Contents'** – Server1.InputBuffer.InProcess is now Server1.InputBuffer.Contents, Server1.Processing.InProcess is now Server1.Processing.Contents.
- **'BatchQueue' queue of an entity renamed to 'BatchMembers'** – So the UnBatch step Quantity default for example now is 'Entity.BatchMembers'.
- **'Node.InputLocation.NumberInProcess' renamed to 'Node.InputLocation.NumberInLocation'**

Seize Behavior

The Seize step was enhanced to dynamically evaluate the quantity (Units Per Object or Number Of Objects). Among other things, this makes it possible to now Seize resources where the desired quantity to seize is in some Min to Max range…by entering the Units Per Object or Number Of Objects in the form 'Math.Max(MinimumDesiredQuantity, Math.Min(Resource.Capacity.Remaining, MaximumDesiredQuantity))'.

New SimBit

SourceServerSinkApproaches.spfx illustrates three different approaches for building a simple model. This is useful for studying the Process approaches versus the standard library approach. It also provides a useful tool for helping students learn to importance of accounting for variability in model data.

Error Checking

We found some common modeling errors coming our way, so we added a bit more robustness to our error checking. If you had previously undetected modeling errors you will need to fix them before running (see Known Anomalies link on Start Page).

Simio Release 3

Simio Release 2 – Sprint 41 – July 21, 2010

This sprint provides enhancements to the standard library (Source) as well as additional engine-level capabilities (Remove) and enhanced output analysis (Subset Selection, Ranking & Selection). In addition you will find the GUI easier to use (Units, Expression Editor, and Display Lists), improved utilization statistics, and even easier animation with the new Node behavior.

Subset Selection
A new button has been provided on the Experiment Design ribbon to help you determine which of a set of active scenarios is the best. If you provide a **Response Objective** (Minimize or Maximize) in an experiment and have run at least 2 replications of each (10 or more recommended), Simio will automatically divide the scenarios into those that are candidates for the best and those that are not for each response.

Enhanced Support for Units
While we have always had support for units on selected properties, we have now extended that support throughout the product.

- Units are now labeled in reports and the pivot grid
- You can change the display units from the Home, Pivot Grid and Reports ribbons
- States now have a "Unit Type" that can be set to indicate the class of items the state deals with (Time, Distance, etc…)
- Numeric properties (like Expression) now have a "Unit Type Property". This can point to a state property, which means "Pull my unit type dynamically from whatever this state property is referencing".
- You can use the Assign step to assign a state a unitized value, depending on the class of the state. For example a Time state can be assigned a value of 100 Minutes
- In an expression, we now take into account a state's unit type. For example, if a Time state (A) is assigned 60 Minutes and you do a delay for "A + 10" (Hours) you will delay for 11 hours
- In the watch window, if a state (or element function) is classified as having units, we show the value (in the base Simio units of hours, meters, meters per hour)
- You can set units on States, Outputs and Tallies

Resource Statistics
Calculating resource utilization has always been problematic because people have different ideas on what it means. We have solved that problem by building in several common definitions as well as providing object builders with the tools to create custom definitions.

- We already had significant information on each resource state including %, time, and occurrence.
- We have refined Units Scheduled and Units Allocated statistics.
- We have added Scheduled Utilization which is the average capacity utilized divided by the average capacity scheduled.

- The standard library objects have clearly defined “Utilized” states such as Busy, Setup, and OffShiftBusy.
- Object builders can implement their own definition of Utilization by specifying if each state should or should not be considered utilized (e.g. perhaps you consider Setup as Not Utilized).

Ranking and Selection Add-in

Often your experimentation will produce multiple scenarios that appear to have roughly similar results, but with differing variability. We supply an experiment add-in named **Select Best Scenario Using KN** based on work by Kim and Nelson that will compare scenarios and iteratively run additional replications to reduce the variability enough that the best alternative can be reliably determined.

Nodes

- **Node Sizes**: No matter what the zoom level, nodes are now always displayed at a size that they can be seen and selected. Previously node sizes changed with the zoom level, sometimes making them to small to easily select or inappropriately large for the model.
- **Node Placement:** The locations of nodes can now be adjusted in the user view of derived objects.
- **Node Parking:** A default parking area is supplied adjacent to each node. Whenever an object (such as a vehicle) parks at the node, if Parking Queue is enabled on the Apperance ribbon, then the object will display on the default parking queue. If you want to customize parking in any way you must disable this option and manually add the ParkingStation.InProcess queue.

Population Member Indexing

It is now possible to directly interact with properties of a specific vehicle or entity. Previously it was difficult to do so unless you had a token for that entity. You can now use a bracketed index to reference directly to a specific member of the population. For example “Truck[3].ResourceState” could be used in a pie chart to show the resource state of the 3rd object named Truck in your system. You can also use expressions for the index and even nest these. This works in State and Event properties, Expressions, and in references on animation objects

- Any agent instance name followed by brackets, with an expression in those brackets giving the index into the agent population, for example:
    - Vehicle[1].SomeFunction
    - Vehicle[1 + 0]. SomeFunction
    - Vehicle[Vehicle[x].SibilingVehicleIndex]. SomeFunction
- They also work for agents inside of other agents:
    - AircraftCarrier[2].FighterPlane[5].Missle[3].TargetIsLocked

Expression Editor

We have improved the expression editor to make it clear when there are additional optional choices available for a function. When you the expression has enough to be considered complete, it will be displayed in bold. When additional choices (required or optional) are available, you will see an indicator that more choices are available.

Drop Down Lists

Any drop down list that could have lots of items now includes a filter at the top that will limit the choices to only those containing the specified characters. For example, typing “server1” into the filter will limit choices to only items involving Server1, or typing “capacity” will limit it to only items involving capacity.

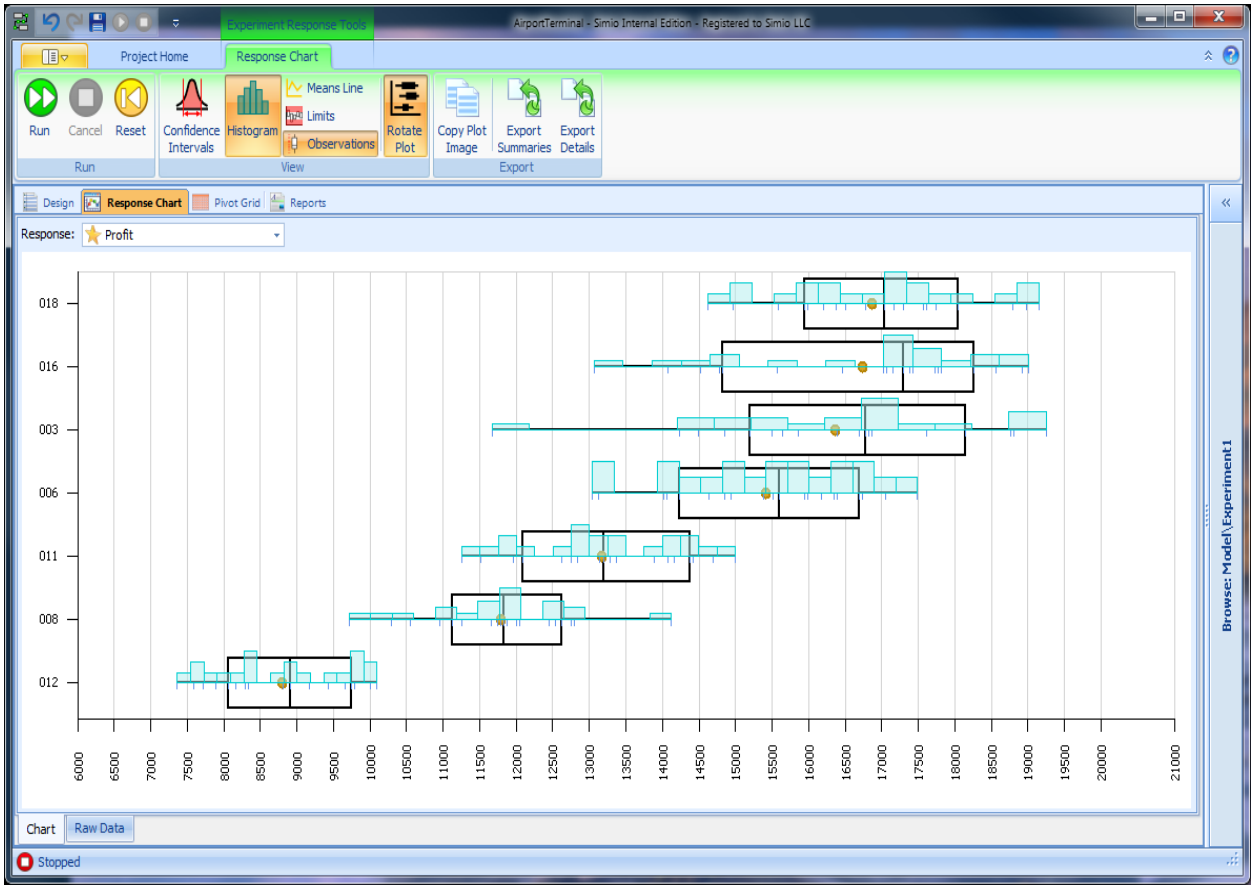
Resized Standard Library Object Symbols

The size of many of the default library symbols have been adjusted to provide a better balance relative to each other and more appropriate to the models that are typically built.

SMORE Plot Enhancements

The SMORE plots help you analyze and compare scenarios.

- **Histogram:** The response chart ribbon now has button to optionally display a histogram. The frequency of the values from the replications is expressed as a histogram superimposed on the side of the box plot to give you additional insight into the data's variability.
- **Rotation Option:** The response chart ribbon now has button to optionally rotate the display by 90 degrees. In certain situations this provides a more effective display.



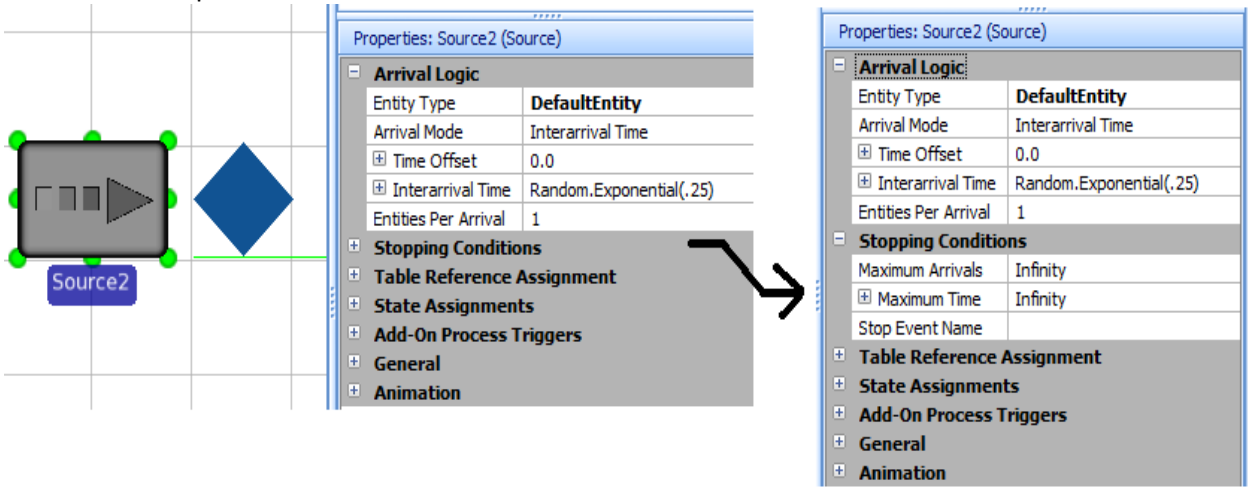
Enhanced Support to Renege From Waiting In A Queue

You may now use the Remove step to remove an object waiting in a queue. This is often necessary to implement a maximum waiting time, preemption from a queue, or other reasons to leave a queue. Removal is supported from the following types of queues:

- Resource Object 'AllocationQueue' (to cancel waiting activity at a Seize step)
- Station Element 'EntryQueue' (to cancel waiting activity at a Transfer 'To Station' step)
- Link Object 'EntryQueue' (to cancel waiting activity at a Transfer 'To Outbound Link' step)
- RoutingGroup Element 'RouteRequestQueue (to cancel waiting activity at a Route step)
- BatchLogic Element 'ParentQueue' and 'MemberQueue' (to cancel waiting activity at a Batch step)

Source Object 'Stopping Conditions'
There is a new 'Stopping Conditions' property category in the Source object that allows you to easily stop arrivals generated by the source using one or more of the following conditions:

- After a specified Maximum Arrivals has occurred.
- After a specified Maximum Time has elapsed.
- After a specified event has occurred.



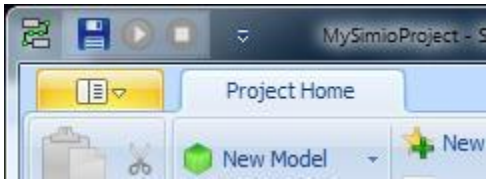
Enhanced Object Behavior

- **Destination Selection**: The queue selection algorithms for the RoutingGroup Element's 'RouteRequestQueue' were updated such that the first eligible route request in the statically ranked route request queue is now selected. Previously, only the first queue item in the queue could be selected as long as it was eligible.
- **Searching**: The Search step can now search Node Lists.

New SimBits

- **RemovingFromAllocationQueue** – demonstrates how the new Remove capability can be used for reneging from an allocation queue.
- **TallyStatisticsInTables** – shows how to use TallyStatistics in combination with tables to collect different types of statistics based on a condition or entity type.
- **VehicleWithVariableRideCapacity** – show how to have a vehicle determine how many entities it will pick up next.

Updated the System Menu and the Applications Menu buttons
Similar to Office 2010, we added a System Menu button (a tiny flow chart) in the far upper left corner and we have changed our large application button to a smaller highlighted menu icon just below it.



Simio Release 2 – Sprint 40 – June 27, 2010

This sprint provides a large number of new features including significant enhancements to the Experiment Window and the Response Charts window, a new Worker object in the standard library, enhancements to Vehicles, Resources, Data Tables, the Source object, enhanced documentation, thirteen new or updated SimBits, and OptQuest evaluation software (Beta) tightly integrated into the Experiment Window.

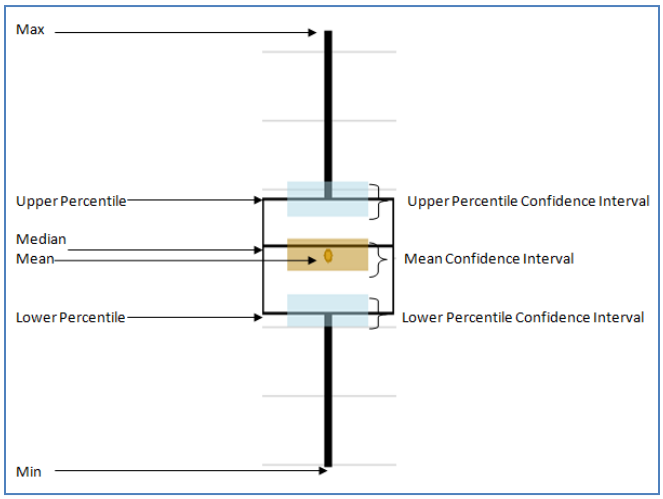
Experiment Window

Two types of Constraints have been added. **Input Constraints** identify expressions that can be evaluated before a run and possibly prevent a scenario from being executed (or sometimes even generated). **Output Constraints** identify expressions that are analyzed after a run and can mark results as invalid if they are out of range. Simio automatically determines if a constraint is an input or output constraint. Ribbon buttons are available to add and remove constraints.

Algorithms by Dr. Barry Nelson have been implemented to assist with automatic **Ranking and Selection** of the best scenarios for a particular response. Often it is hard to determine a clear "best" with statistical rigor. In that case the group of scenarios which contain the best scenarios will be distinguished from those that are not the best. This same algorithm is used in the response charts (see below).

Response Charts

We have taken advantage of state-of-the-art research in output analysis to build in a graphical tool to help you analyze and compare scenarios. We have extended recent developments by Dr. Barry Nelson to generate **SMORE Plots** (**S**imio Measure **o**f **R**isk & **E**rror) to display both the expected value of a scenario and multiple levels of variability behind the expected value.



The chart features:

- Maximum Value
- Upper Percentile – User definable
- Confidence Interval around Upper Percentile
- Mean
- Confidence Interval around Mean
- Median
- Lower Percentile – User definable
- Confidence Interval for Lower Percentile
- Minimum Value

A **Response Chart Ribbon** allows you to enable/disable specific graphic features: confidence intervals, individual observations, upper/lower bounds, and connecting lines between means of scenarios.

A **Raw Data Tab** allows you to view the full analysis data in a tabular format.

Worker Object

A **Worker Object** was added to the Standard Library to make it easier to model typical activities associated with people. A Worker may be used as a moveable resource that is seized and released for process tasks in the system. Additionally, the worker object may be used as a transporter to transport entities between node locations. Workers can follow shift schedules and they have a full set of states to allow tracking and analysis of their activities.

Each Worker object instance defines a *dynamic population of moveable unit resources* in the modeled system. Each individual worker in the population may have a resource capacity of 0 (Off Shift) or 1 (On Shift). A worker/vehicle can follow an On Shift/Off Shift work schedule or always be available (i.e., have a 'Fixed' capacity of 1).

Use the Seize and Release steps to seize or release a Worker for a process task. Upon seizing a worker object, you may use the 'Request Visit' option in the Seize step (Advanced Options) to optionally require the seized worker to visit a specified node location before the seize is considered 'completed'.

Refer to the two new Worker-related SimBits: 'WorkerUsesWorkSchedule' and 'WorkerUsedForMultipleTasks'

Vehicle Object

We have added both major and minor enhancements to the **Vehicle** object. Within the properties window many of the properties have been relabeled and reorganized for both clarity and consistency with other similar objects. The somewhat confusing Idle Action of 'None' was changed to the clearer 'Remain In Place'. Two new add-on process triggers were added ('Allocated' and 'Released') to allow better control and the 'Allocated' and 'Released' events fired by the vehicle were made public. We have added new states and state tracking to provide better analysis of vehicle activities. The 'ResourceState' state variable for the Vehicle object now tracks the possible state values:

- **Idle** – The vehicle resource is not allocated to any task.
- **Busy** – The vehicle resource has been seized by another object in the system for non-transport related tasks.
- **Transporting** – The vehicle resource has been allocated to a set of transport tasks.
- **Failed** – The Vehicle is failed and unable to move or perform transport pickups/drop-offs. Any transport or non-transport related processing using the vehicle resource has been assumed suspended or blocked.
- **FailedBusy** – The vehicle has failed while seized by some other object in the system for a non-transport related task, but the task is assumed unaffected by the vehicle failure because the vehicle has already been moved to the task's desired location.

Source Object

We have added a new Table Reference Assignment category to the standard library source object. This allow you to associate a particular row and table with an entity that is about to be created or an entity that has just been created, without having to use process logic. This also makes it easier to create entities where the distribution of entity types is specified in a table. See the SimBit 'SelectEntityTypeFromTable.spfx' for an example.

Resource Object

We have added new states and state tracking to provide better analysis of resource activities. The ResourceState now has the following possible state values:
Idle – The resource is not allocated to any task.
Busy – The resource has capacity allocated to one or more tasks.
Failed – The resource is failed and not allocated to any tasks.
OffShift – The resource is off shift and not allocated to any tasks.
FailedBusy – The resource is failed but allocated to one or more tasks and still assumed 'Busy' though failed.
OffShiftBusy - The resource is off shift but allocated to one or more tasks and still assumed 'Busy' though off shift.

The 'FailedBusy' and 'OffShiftBusy' states are new. At the beginning of a failure or off-shift period for the **Resource** object, if the resource is allocated to process tasks, then by default the resource is now considered 'Busy' during that time and its state is set to 'FailedBusy' or 'OffShiftBusy' respectively. The downtime period begins immediately, and no new seizes of the resource may take place, but by default any processes out in the system utilizing the resource are assumed to not be affected and will finish their processing. Refer to the new SimBit 'ResourceStateWhenOffShift'.

To interrupt processes utilizing a resource when a downtime period begins, you may add your own custom logic using tools such as the **Interrupt** step to preempt and release the resource during the downtime period.

SimBits

The following new SimBits were added:

- ExamplesOfFunctions_StaticObjects
- ExamplesOfFunctions_DynamicObjects
- UsingAStorageQueue
- ResourceStateWhenOffShift
- AddAndRemoveCapacity
- MovementInFreeSpace_WithATargetLocation
- WorkerUsesWorkSchedule
- WorkerUsedForMultipleTasks

The following SimBits were updated to illustrate new techniques:

- EntityFollowsSequenceWithTable
- EntityFollowsSequenceWithTable2
- InterruptingServerWithMultipleCapacity
- SelectEntityTypeFromTable
- SortingConveyorSystem

ModelEntity Object

We made some minor changes within the properties window to improve both clarity and consistency with other similar objects.

Tables

When using the relational data tables feature, we have added a new viewing option, a **Master-Detail View**. For things with "detail" views (collections of rows pointing to a specific keyed row in a table), you'll see a little plus sign at the front of the row you can use to see the detail view. The detail view essentially allows you to view the associated rows of the related table. The detail view can be edited just like the main view (insert, remove rows, edit rows, copy/paste, etc…).

New Help Topics

In addition to continually enhancing our help and reference materials to describe new and revised features, we are continually improving our general content to better explain concepts that users have found confusing. Here are some new topics added to help better explain the available functions, states and events:

- Functions, States and Events for Entity Objects
- Functions, States and Events for Objects Enabled as Resources
- Functions, States and Events for Node Objects
- Functions, States and Events for Link Objects
- Functions, States and Events for Transporter Objects

Also please refer to the two new SimBits: 'ExamplesOfFunctions_StaticObjects' and 'ExamplesOfFunctions_DynamicObjects'.

User Interface

A new keyboard shortcut ('Ctrl' + '+') was put in to insert rows into any data grid where items can be added (e.g. things like tables and lists).

Resized Entity Object Symbols

The size of the default entity has been reduced to better reflect sizes of typical entities.

OptQuest (Beta)

OptQuest is offered for Simio as an optional add-on to the standard Simio products. OptQuest enhances the analysis capabilities of Simio by searching for optimal solutions to your models. OptQuest helps remove some of the complexity by automatically searching for the optimal solution for you. You describe your simulation problem and it searches for input controls to maximize or minimize your objective (i.e. maximizing profit or minimizing cost).

Although the full use of OptQuest for commercial use requires activation, the included evaluation mode allows up to two controls and two constraints, and will produce a up to 20 scenarios.

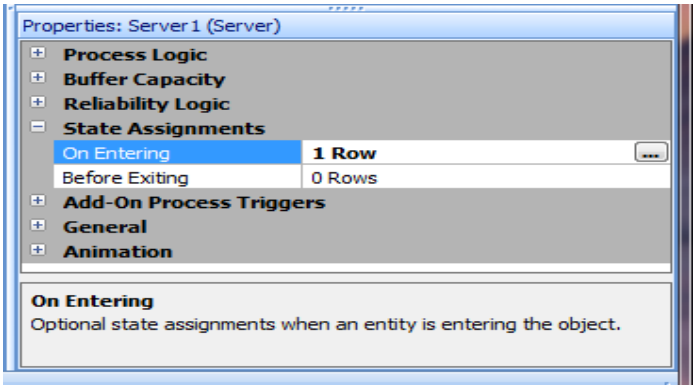
Look in the Experiment Window of the AirportTerminal example model for an example of applying OptQuest to a realistic project.

Simio Release 2 – Sprint 39 – June 6, 2010

This sprint provides new capability to assign states without using processes, interruption and preemption of processes, more effective choices for selecting destinations, and many new features added to the experiment window.

New 'State Assignments' Property Category for Standard Library Objects

For many standard library objects there is now a **State Assignments property category** where you can easily specify the assignment of state values when an entity is entering an object, or when an entity is ready to exit an object. This feature makes it much easier to do common state assignments without having to use add-on processes and manually place Assign steps.



Advanced Capability for Process Interruption

The **Interrupt step** may be used to interrupt process delays. A token delaying at a process' Delay step is considered a candidate for interruption if the Interruptible property of the Delay step evaluates to True. To add logic for handling interrupted items (e.g., re-routing logic), the Interrupt step can create new tokens that are associated with the interrupted items. These tokens will exit the 'Interrupted' exit point of the step.
The Interrupt step may be used to model resource preemption scenarios, allowing model logic to interrupt a process delay activity that is using a resource, in order to release the resource for a higher priority activity. This approach allows very flexible preemption capability with precise control over which entities are preempted and how the preempted entities are subsequently processed.

New SimBits

InterruptibleOperator - demonstrates how to interrupt a moveable operator to do more important work at a different location.
InterruptingServerWithMultipleCapacity – demonstrates how to preempt a resource away from low priority work and switch to higher priority work.
InterruptingAcrossMultipleServers – demonstrates how to interrupt processing at a one of several servers with a lower priority item to instead process a higher priority item.

InProcess Ranking Rule property on Station element

The Station element now allows you to specify the ranking rule of the station's InProcess queue that holds all of the entity objects physically located in the station.

The Standard Library objects were updated such all InputBuffer stations have the same InProcess Ranking Rule as the resource capacity allocation rule being used. Thus, for example, if entities are waiting to seize a Server object with a static ranking rule of Lowest Value First (Entity.Priority), the InputBuffer.InProcess queue where the entities are physically waiting will also be ranked Lowest Value First (Entity.Priority).

New InputLocation Functions For Nodes

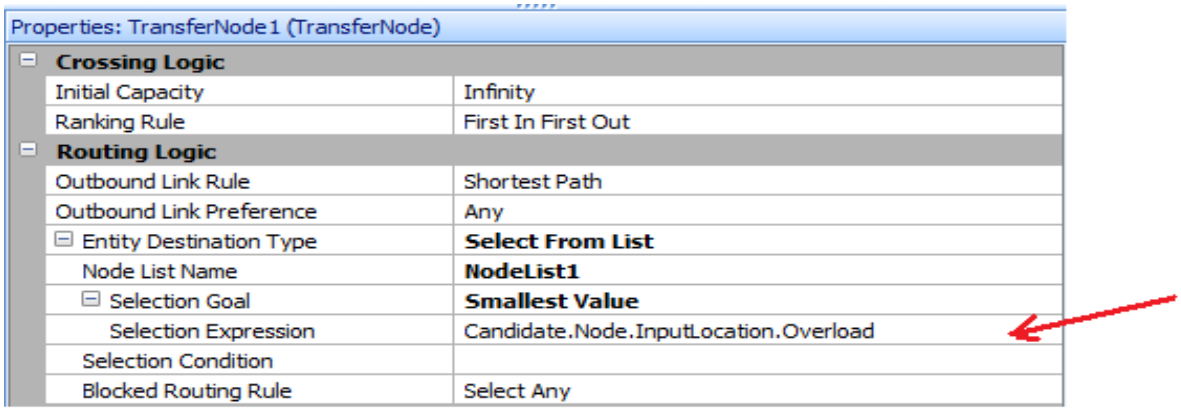
For any external input node, a new set of **InputLocation functions** are now provided to return some basic information on the node's input location, such as the input location's capacity, remaining capacity, load, or overload. Such information on the input locations of input nodes is often useful or required to implement dynamic routing logic. For example, in the TransferNode object in the Standard Library, when setting an entity's next destination using the Select From List option, the new functions are useful in selecting the best destination node from a list of candidates.

Six new functions are provided:

- InputLocation.Capacity - returns the current capacity of the location inside the associated object.
- InputLocation.CapacityRemaining – returns the current available (unused) capacity of the location inside the associated object.
- InputLocation.NumberInProcess – returns the current number of entities in the location.
- InputLocation.NumberWaitingEntry – returns the current number of entities waiting to enter the location.
- InputLocation.Load – returns the "load" which is the sum of entities on their way to the node, waiting to enter the object, and now in the object.
- InputLocation.Overload – returns how much a location is overloaded (a positive value) or underloaded (a negative value) or more specifically Load minus Capacity.

TransferNode Object Changes

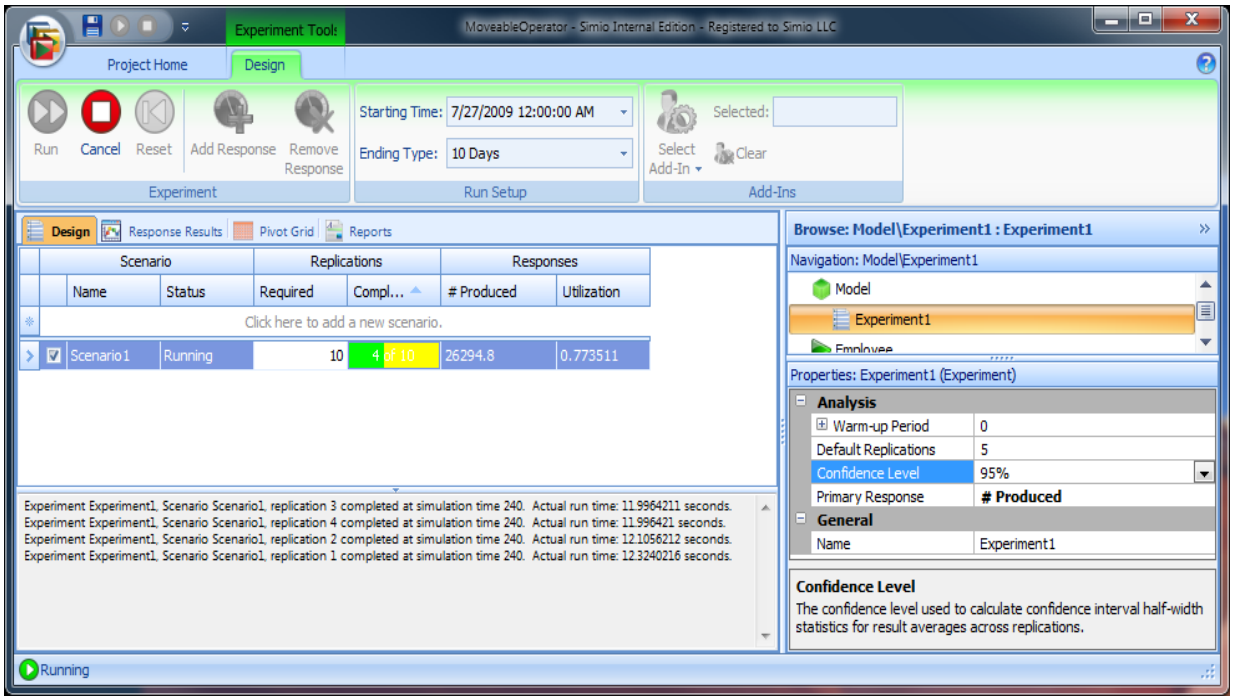
Related to the above change, the default value of the destination *Selection Expression* when Select From List is now Candidate.Node.InputLocation.Overload. In many cases a selection of the smallest Overload will provide the server most capable to handle another entity because it takes into account many different aspects of actual and potential work load.



Experiment Changes

We are adding capability to the Experiments window to improve its usefulness as an analysis tool as well as to make it easier to use with various add-ins like OptQuest. Here are some of the latest changes:

- Columns are categorized for easier reading. Columns and categories can be reordered.
- Columns may be sorted and filtered. If you sort on a response, the rows (scenarios) will be dynamically resorted by your sort criteria. Multiple sorts can be added with shift-click.
- A checkbox has been added to the left of each scenario. Only scenarios that are checked will be enabled. This provides a way to do additional replications on selected scenarios.
- 'Completed' column now provides numerical and graphical feedback as the replications progress.
- Responses now have Lower Bound and Upper Bound properties used for highlighting unacceptable scenario results and for use with user add-ons.
- Responses now have an Objective property that will be used with ranking and with user add-ons.
- There is now a Primary Response property on the experiment that is available for use with user add-ins (typically for defining an overall objective function).

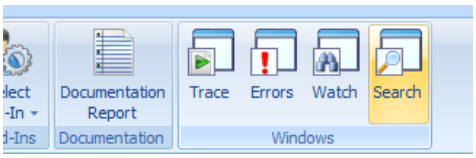


Simio Release 2 – Sprint 38 – May 15, 2010

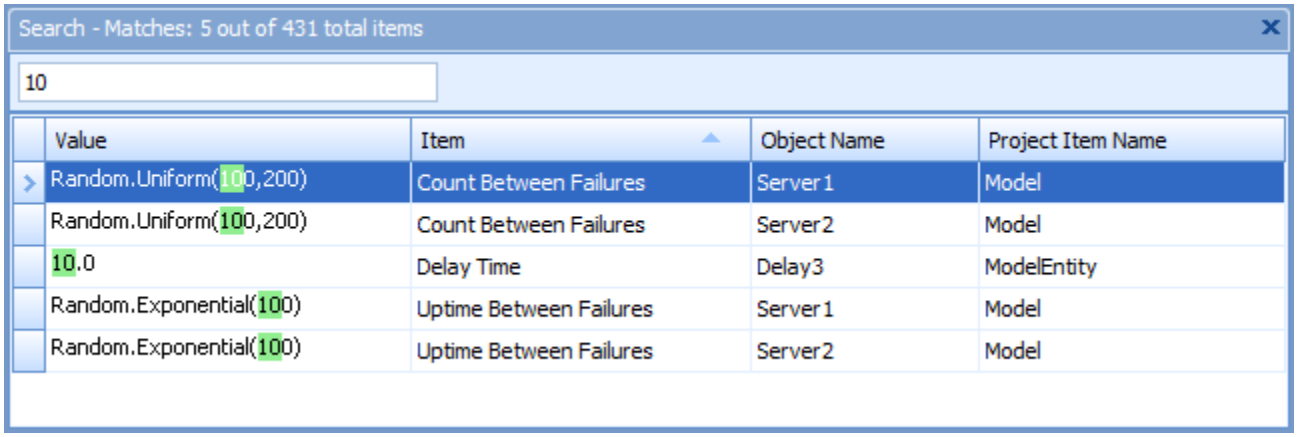
This sprint provides new capability to search for items in your model, additional multi-select features, enhanced ability to graphically compare experiment scenarios, new modeling capability, and easier installation options.

Search Capability

There's a new button on the "Project Home" tab which shows and hides a new Search window at the bottom of the application.



When you type something into the Search window's text box, the window will display everything that it finds that contains that text.



Double-clicking on a row in the results will generally take you to the window containing the item where the text is found and (if possible) select the property where the text is found. The columns can be manipulated to sort, rearrange, and filter.

New 'Location' Functions on IntelligentObject

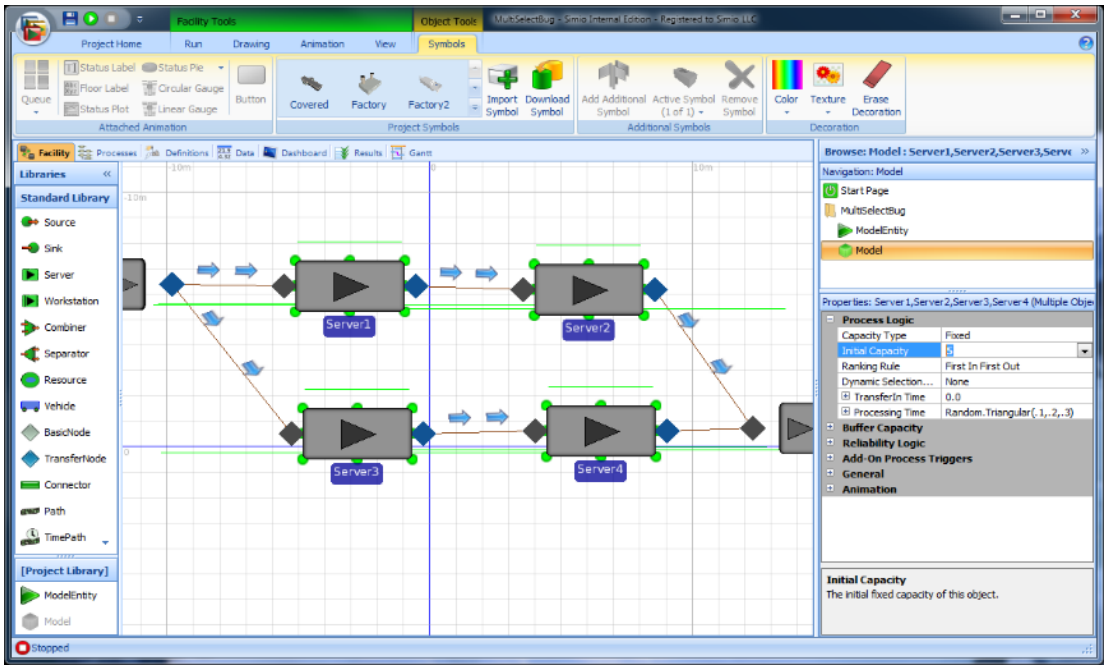
Location.X, Location.Y, and Location.Z functions have been added to return the X, Y and Z coordinates (location) of any object. For static objects it generally returns the XYZ of the object instance or its starting node. For dynamic objects like entities, it will return the directed second order Movement state or a value dependent on an object associated with the entity.

Multi-select

The multi-select feature for the property grid is now available. You can select multiple items and the intersection of their set of properties will show up in the property grid. If they share values for a property, the value will show up, otherwise the value will be blank. You will be able to do things like select multiple links and turn them all bidirectional, or select multiple output nodes and have them all use the same transporter.

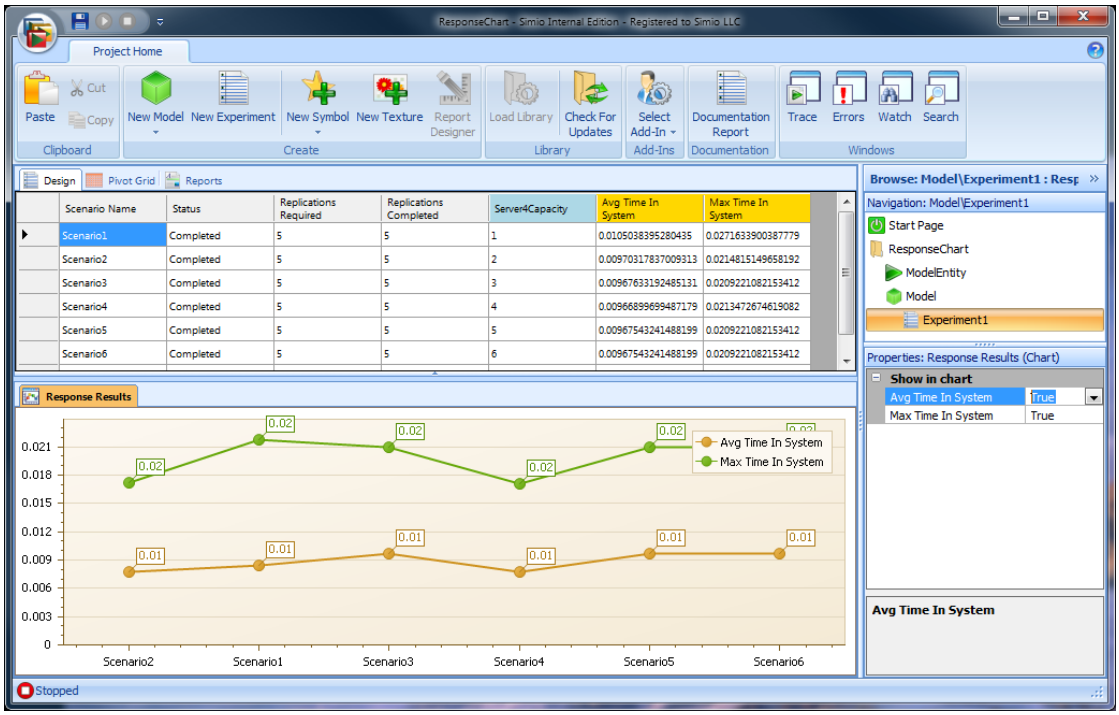
Faster API

In Simio's programmatic API, there is a new method on the IModel interface called BulkUpdate, which can be used to greatly improve performance when adding a large number of objects to a model.



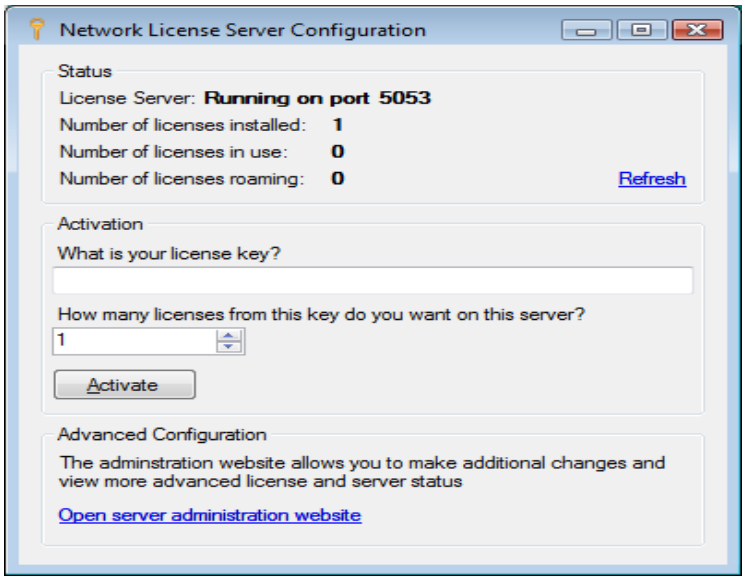
Experiment Graphics

You can now see and compare your **experiment response variables**. The Response Results Tab creates a graph using the average value for each response of each scenario, essentially graphing the Response columns from the design window, in real time. This graph simultaneously graphs the outputs of all responses in the experiment on the same Y-axis scale. You can control what responses are displayed in the graph by selecting either True or False from the drop down listed next to each response in the Show in Chart window. When you change what responses are displayed in the graph, the Y-axis automatically scales itself to properly display the data.



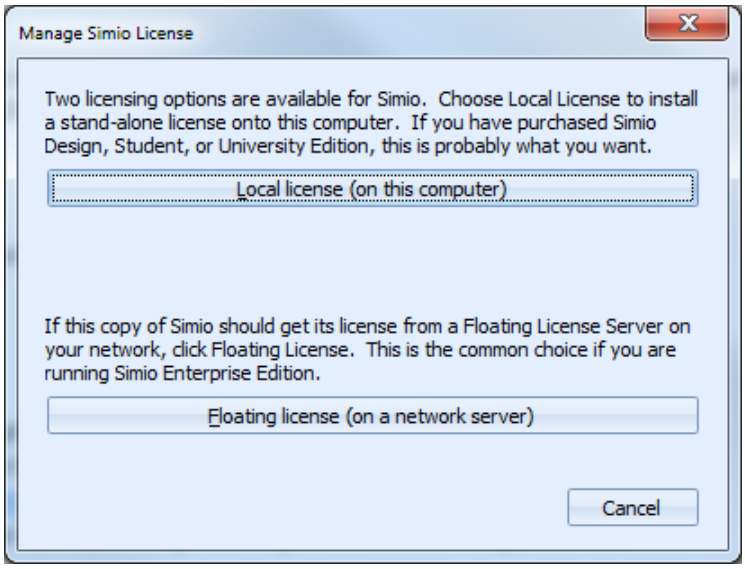
Server Activation Installer

This new installer for the Simio Enterprise Edition activation server will install the license server, set it to start up as a windows service, and adjust the standard windows firewall (5053 and 5054) to allow for client connections. If users are running a different firewall, they might have to open these ports manually. It also installs a configuration utility that makes it easy to activate keys:



Activation Switching

On upgraded Simio installations you can now more easily convert between local and server-based licensing. The first time you click the licensing button (or link on the top right), you will get this dialog asking what you want to use:



If at any time you want to switch the type, both licensing dialogs now have links at the bottom which will also bring up the above dialog.

Simio Release 2 – Sprint 37 – April 25, 2010

This sprint provides the long-awaited multi-select feature. It also contains a new book which we expect you will find very helpful in learning more about Simio. Don't miss the enhancements to the standard library objects as well.

Multi-select
Multi-select capability is now supported in the graphical views (i.e. Facility, Dashboard and Eternal View windows).

Ctrl+click on any object adds (or removes) it to the selection set. (Ctrl+click on a node no longer starts drawing a link, you now must use Ctrl+Shift+click.) Ctrl+click and drag in free space will do a box select of objects within the box.

Once a group of items is selected, you can move, copy or delete the entire group of objects in a single action.

New Learning Materials

We are now including a new book called ***Introduction to Simio***. This short, six-chapter book walks you through many product features and is an excellent way to get to know Simio. The existing ***Simio Reference Guide*** can be used as a supplement for looking up detailed feature descriptions. You can find a link to both books on the Simio Start Page.

The Simio Start Page has been redesigned to make it easier to find and take advantage of the many important resources available.

Source Object Enhancements (Standard Library)

The Source object was updated to provide an **Initial Number of Entities** option for the 'On Event' arrival mode, for creating an initial number of entities at the beginning of the simulation run in addition to entities created by the specified Event Name. This makes it very easy to "seed" a model with initial entities and then replenish the supply with event-based creations.

A **Rate Scale Factor** option was added for the Time Varying Arrival Rate arrival mode to allow easy scaling of the arrival rate values specified in the rate table. Once you have an appropriate arrival pattern, this feature makes it easy to adjust the overall arrival rate by a factor (e.g. a 10% increase or decrease).

Resource Object Capacity Now User-Assignable

Every resource object (e.g., a Server object in the Standard Library) now has a **CurrentCapacity** state that may be assigned using an Assign step to dynamically change the object's capacity. This can be used to create more flexible schedules as well as logic-based capacity changes.

Node Object Function Changes

NumberRoutingIn, NumberRoutingIn.CanEnterObjects, and NumberRoutingIn.Transporters functions were added. These functions allow you to determine the number of entities that have a particular node set as their destination, as well as which of those entities can actually enter the associated object of the node (refer to the 'Can Enter Objects' property available on an entity type) or are transporters. These functions are particularly useful in destination selection logic.

Selection Expression Capabilities

We have exposed new capability to use when selecting a destination on a Node. In addition to looking at properties of the candidate nodes, you can also consider properties of the associated obects of those nodes. For example, your selection expression might include a term like Candidate.Server.Capacity when all your nodes have an associated object of type Server.

Improved Results Classification for OutputStatistic, TallyStatistic, and StateStatistic

The OutputStatistic element, TallyStatistic element, and StateStatistic element now provide more user control for classifying the statistical output in the model results (pivot grid or reports). Optional user-defined strings may now be specified for the Data Source, Category, and Data Item fields

File Size of Release Image

When you download this and later releases you should find the download to be noticeably faster. We have reduced the file size by more than 40%.

New SimBit

SimpleLeastSlackSelectionRule – Entities with different due dates are selected for processing order using a Least Slack scheduling methodology .

Simio Release 2 – Sprint 36 – April 4, 2010

This sprint provides significant new vehicle features, enhanced user code capabilities, and more. And we continued work enhancing the Simio API, architecture, and general software robustness.

Vehicle as a Resource

The Vehicle object now has a 'ResourceState' state variable that tracks its current state (Idle=0, Busy=1, Failed=3). Statistics on this resource state are now automatically provided in the results, and the resource state variable may also be easily referenced for logical or animation purposes.

An 'On Demand' Vehicle object may now also be used as a moveable resource that is seized and released by model logic for non-transport tasks. To prioritize specifically among transport tasks, you continue to have the 'Task Selection Strategy' available in the 'Transport Logic' category of the Vehicle object's properties. To prioritize specifically among non-transport tasks, you may use the Ranking Rule or Dynamic Selection Rule properties available in the 'Process Logic' category of the Vehicle object's properties.

If you want to implement even more customized control of how an 'On Demand' Vehicle object selects its next task…then you may take advantage of the Evaluating Rider and Evaluating Allocation add-on process triggers. For example, you can use those add-on processes to implement logic on a particular vehicle such as "Don't accept any non-transport tasks until all my transport tasks are finished first".

Please refer to important compatibility notes in Known Anomalies (See Simio Start Page)

Files Element

Two changes have been made to make the Files Element (and the associate Read/Write Steps) easier to use.

- Previously, you had to always specify the full directory path for the file. Now, if the user does not include the directory name in the File Path property of the element, then the directory of the Simio project is assumed.
- Previously, writing to a file from an experiment raised an error when running multiple processors because you had multiple execution threads trying to open the same file. Now, if using the Write step to write to a file during an experiment run, multiple files are now created and the text '_[ExperimentName]_[ScenarioName]_Rep[ReplicationNumber]' is appended to each file name to make them unique.

User Extensions Templates & Documentation

If you are interested in extending Simio with your own custom steps, elements, and rules, it just became a lot easier. In addition to over 400 pages of programmer-specific documentation, we have just added some overview instructions to make it easier to get started. You will find that information in both the on-line help and the Simio Reference Guide.

Better still, we have also added Visual Studio project and project item templates for common tasks. When you are creating a new project, using these templates will give you a fully structured code skeleton that lets you immediately create a sample. So you can spend more time implementing your features and less time working on the “plumbing”.

Graphics Engine

Some systems have had nagging problems related to graphics and especially interaction with screen savers. We have implemented major updates and revisions to our internal graphics handling to try to eliminate these problems. If you still have problems, please contact us with details so we can focus on your specific hardware problems.

Five New SimBits

SeizingVehicle - Model a system that has a vehicle which is used for both transport and non transport activities. The vehicle’s capacity is seized by another object for the non transport activities.
SearchTables - Search through a Data Table and select a row based on certain criteria.
LeveledArrivals - Model a system where parts of different types arrive in a certain order.
EntityMovementInFreeSpace - Control the movement and direction that my entity moves in Free Space.
CONWIP - Model a CONWIP system (constant level of WIP). The system should begin with 10 entities in the system and as soon as one entity leaves the system, another should arrive.

Simio Release 2 – Sprint 35 – March 14, 2010

This sprint provides a few new features, but was again highly focused on architectural changes and robustness. It continues work enhancing the Simio API and solves situations found in our internal release related to the changes to file formats.

Copy/Paste now supports copying spreadsheets to/from many new constructs.

Search step now allows searching of table rows in addition to the various lists of objects. In addition to the general purpose benefits, this makes initialization of WIP in objects easier to do.

Startup modes allow handling of situations when you have trouble getting Simio to work correctly on a new system. Such problems are usually due to problem with the machine's graphics hardware and software. Starting in **Safe Mode** disables graphics acceleration in hardware. Starting in **OpenGL Mode** changes from DirectX to OpenGL to work around some systems that have limited DirectX support. While these offer help in working around computer problems, we recommend consulting our on-line knowledgebase for other tips on solving the problems.

In the 'General' category properties of an object, there is now a new **Size And Location** category. A new **Current Size Index** feature is now also provided. This property specifies the optional name of a state variable that indexes into the object's associated symbol list, and whose value will automatically resize the object to match the size of the indexed symbol.

The **Quantity Available** state of a **Material** element may now be directly assigned using the **Assign** step, as an alternative to using the **Produce** and **Consume** steps to produce or consume available material quantities.

The **Pivot Table** has been enhanced to display the words Infinity, Correlated, Insufficient, and NaN (Not a Number) when appropriate based on the observations and the calculations being performed.

Floor Labels can now be drawn inside a symbol to make it easy to label different symbols that may change during a run.

You can now reorder properties within a list of property definitions in an object by selecting **Move Up/Move Down** from the ribbon.

We added 3 **SimBits**:
UsingButtonsToAffectSystem illustrates how to add buttons to your facility view to allow easy user interaction and control during a run.
DefineEntityProperties shows how to define and use additional custom properties on an entity.
WorkstationWithMaterialConsumptionAndReplenish illustrates how to implement automatic material replenishment policies in a model that consumes materials.

Simio Release 2 – Sprint 34 – February 20, 2010

This sprint answers a series of frequent user requests in addition to continuing work on the API and internal changes which will eventually result in additional new features.

Time Units Display
A Time Units field has been added to the Animation section of the Run ribbon. The value of this is used to set the numeric format and the labeling on both the elapsed time on the status bar and the time values displayed in the trace window.

Background Color
The Background Color button on the View ribbon allows changing the background color of the entire Facility window floor in both the 2D and 3D views.

Documentation Report
A Documentation Report button has been added to the Project Home ribbon. This generates an html report containing a summary of all objects and the non-default data and properties in your model. This is useful for capturing a snapshot of your model for examination or storage outside of Simio.

Object Updating
When objects are changed within a library, you receive the option to adopt or ignore those updates. You will now have the option to select or ignore the update for each individual object. And if you choose to ignore any updates, a **Check for Updates** button has been added on the Project Home ribbon to allow you to accept that update at a later time.

Finer Event Calendar Control
We have added a new function named **Math.Epsilon** that returns the smallest positive real value. This can be useful when you want to have a selection or expression that involves a very small value or increment so your logic can discern the difference.
We have added new Delay step behavior with a special interpretation of Math.Epsilon to mean "delay by 0.0, but put the entity at the end of the current event list" so it will execute after the end of all other events at the current time (also known as a "late event").
We used the above Delay behavior to fine tune event timing on Nodes. Now, after an entity moves out of a node, the selection of the next item (if there is a choice) is treated as a late event to allow all other entities to move to a position where they are eligible for selection.

Entity Movement
We have improved the implementation and documentation of **Heading**, **Pitch**, and **Acceleration**. This allows control of how entities move through free space. See the help topic Entity Routing and Movement for full details.

New file format
Project files can now be saved in a format that is significantly compressed. The new default extension for saving is **spfx**. New projects will save in this format unless you change the extension to **spf** (the old format). Existing files will keep their current extension when you click "Save", so old files will continue to save as spf unless you "Save As" and specify the extension spfx.

Automation/Programming API
An application programming interface (API) is an interface implemented by a software program to enable interaction with other software. Simio has draft API for defining and running experiments and editing table data and retrieving results. It is still incomplete and only lightly documented, but adventuresome souls can get a sneak peek at the evolving API.

Initial Location of Agents/Entities/Transporters
An object created by the *InitialNumberInSystem* or **Create** step (CreateType=>'NewObject') is now initially located in free-space at the object instance location.
An object created by the **Create** step (CreateType=>'CopyParentObject' or 'CopyAssociatedObject') is now initially located in free-space at the same location as the original object that was copied.

Transfer Step Changes
Transfer now allows entity transfers between object locations without the entity using external nodes to enter and exit objects. So, for example, you can immediately transfer an entity from inside object A to a node outside object B using the Transfer step. This is an advanced feature that should be used with caution because this action could conflict with other related actions taking place in an object (such as other tokens from that entity may still be in process).

SimBits
We added 3 SimBits:
EntityStopsOnLink illustrates how to make an entity stop on a link based on an arbitrary condition.
OneQueueForMultipleServers shows how to feed multiple servers from a single queue.
SortingConveyorSystem illustrates a conveyor that sorts boxes by destination.

Simio Release 2 – Sprint 33 – January 30, 2010

This sprint we did some minor enhancements and bug fixes, plus a great deal of work in two large areas which will require additional work before they are fully usable.

View Step Parameters in Processes

If you hover the mouse over any step, a pop-up will show all parameters that do not have their default values. This, in combination with the Process Description, Step Name, and Step Description, makes it much easier to quickly understand what a process is doing and how it works.

Confidence Level and Batch Means

The confidence level is used to calculate confidence interval half-width statistics for result averages across replications or batches.

- Added a default confidence level to experiments so you can select the confidence level appropriate to your experimentation. Also added a default number of replications.
- Changed TimeInSystem tally on the Sink object to use default confidence level.
- Now, whenever a single replication is run in an experiment, all TallyStatistics and StateStatistics defined by the user or defined in the library (e.g. TimeInSystem on the Sink) will use an internal batch means technique to calculate a confidence level that is reported as an additional row of data (Half Width) for that statistic. Multiple replications will continue to calculate a cross-replication confidence interval (Half-Width) in a separate column.

Copy and Paste

The ability to Copy and Paste has been added to almost every construct. More related enhancements such as being able to select multiple objects before copying are still in process.

Automation/Programming API

An application programming interface (API) is an interface implemented by a software program to enable interaction with other software. Simio has had a small and growing API for quite a while. We have just started on a major project to expand and complete this interface. It is still incomplete and only lightly documented, but adventuresome souls can get a sneak peek at the evolving API.
When complete this will add significant capabilities for interfacing Simio to other programs for data exchange, optimization, Simio controlling or being controlled by another program, and lots more.

Simio Release 2 – Sprint 32 – January 9, 2010

In this sprint we have made several changes to improve modeling capability, on-screen and reported statistics, and execution speed.

List States can now associate state names with auto-states

If a list state is being automatically assigned by the simulation engine, you can now define extra possible state names for the list state and associate those names with auto-states. You can then also manually assign values to the auto-assigned list state in process logic using an **Assign** step, and the assignment is allowed as long as the new state to be assigned is associated with the current auto-state condition. The combination of these two features allows you to define sub-states to an internally defined auto-sate, for example defining Setup, Turning, and Teardown states that are all associated with the Busy auto-state.

Workstation object (Standard Library) 'ResourceState' enhancement

The **Workstation** object now has a *ResourceState* list state (similar to that provided by the Server, Combiner, and Separator objects) with the following possible state values: Starved = 0, Processing = 1, Blocked = 2, Failed = 3, OffShift = 4, Setup = 5, Teardown = 6.

The *ResourceState* may be referenced for process logic purposes, and also provides state-based statistics results.

Link.NumberTravelers.Accumulated Functions And Results

The following functions are now provided for any link:

NumberTravelers.Accumulated
NumberTravelers.Accumulated.Minimum
NumberTravelers.Accumulated.Maximum
NumberTravelers.Accumulated.Average

An entity is counted as an 'Accumulated' traveler on a link if:

a) The entity has reached the end of the link and has been stopped there **without being engaged to the link**.
b) The entity's front edge has collided with the back edge of an entity in front of it on the link, and the entity has accumulated behind that entity **without being engaged to the link**.

Once flagged as 'Accumulated', the entity will continue to be considered 'Accumulated' until either its front edge leaves the link or the collision situation is cleared.

Accumulation-related statistics are now reported in the Results. Accumulation statistics are particularly useful when using accumulating conveyors (**Conveyor** object in standard library).

Connector object (Standard Library) change

The **Connector** object in the standard library is a link that is hard-coded with its *DrawnToScale* property set to 'False' and its *LogicalLength* property set to '0.0'.

The Simio engine now consistently has special treatment for a 'connector' link configured as above such that entities will never actually enter, move across, and exit such links…but will always instead immediately do a node-to-node direct transfer across the link. This behavior optimizes run performance using such links.

Cut, Copy and Paste

We are steadily enhancing Simio's cut, copy, and paste capabilities. In this release support has been extended to include:

- Elements
- Properties
- States
- Events
- Lists
- Tokens
- External Nodes (in the external view)
- Tables
- Function Tables
- Rate Tables
- Schedules
- Changeovers

Simio Release 2

Sprints 29-31 – December 12, 2009

We are very pleased to bring you Simio Release 2. If you have not seen Simio lately, we think you will be surprised by the progress we have made. We have left the "new product" phase behind and dramatically improved both the feature set and the robustness of Simio. The next 14 pages describe the plethora of new features that have been added since we shipped Simio 1.0 just 7 months ago. Simio is now truly a world-class product ready to take on your most demanding modeling tasks.

If you have been updating regularly to the latest sprint, I think you will still be impressed with what has been added since Sprint 28. We listened to your feedback. We have made UI changes throughout the product which will make Simio much easier to use. We have also added relational database capability, features to make it easy to model movable operators, network server licensing, and much more. You can find more details in the next 3 pages.

Relational Tables

We have added the capability to define multiple tables which can automatically reference each other. You will see two new buttons in the Table view of the Data window:

- "**Set Column as Key**" – After selecting a column you can click this to mark it as a "Key". This means all values in the column must be unique across all rows in the table.
- "**Foreign Key**" – Adds a ForeignKey property to the table. This is how you create a relation to another table. Once added, and selecting the column, you can set "Table Key" property in the property grid to a "Key" in another table.

We also added the **TableName.AvailableRowCount** function, which returns the number of available rows for the table given the current table selection context.

Two new SimBits were added to illustrate how the new relational table features work.

- **UsingRelationalTables** – An example of setting up multiple tables that are used together.
- **EntityFollowsSequenceWithTable2** – Illustrates the same functionality as the similarly named SimBit but using relational tables.

Moveable Operator

- A **RequestVisit** option has been added to the Seize step and the Workstation object 'Secondary Resources' section to allow requiring a moveable resource to visit a specified node for use during a seize.
- An **Arrive step** has been added for use in building a 'moveable resource' object. This allows an object (the moveable resource) to notify that it has 'arrived' to a requested visit location.
- Updated the **MoveableOperator SimBit** to use the new features. This now illustrates a simple moveable resource object that can be seized, then requested to come to a location, and the token exiting the Seize step and continuing processing when that requested resource object has 'arrived'.
- Added a **New Model -> Entity Model -> Operator** option to make it easy to create an object that will be used as a moveable operator.

SimBits & Examples

- **VehicleVisitsServiceCenter** illustrates a customized transporter that completes its current move and then goes to a service area when it senses a pending failure condition.
- **AirportTerminal** illustrates an example of airport ticketing, check-in, security, and boarding.
- **ManufacturingAssembly** illustrates the assembly and shipping of lamps, including lift trucks, conveyors, and moving resources to move parts.

Network Licensing

We have added the capability for Enterprise Edition and University Edition activation to be installed on a network server. This allows one or more licenses to be shared across multiple users and even multiple geographic locations. The server also supports “borrow” capability so that a network license may be temporarily moved to a local machine and used independent of any connection to the server.

Vehicle

The Vehicle now has a new **Task Selection Strategy** named ‘First In Queue’. This is also the new default strategy because it is the fastest. When a Vehicle object searches a RidePickupQueue, the global VisitRequestQueue, or its RideStation this option will simply select the first eligible waiting rider in the queue without a full dynamic search through all queue members.

Major UI Simplification

We made a series of changes to make **navigation** around your model significantly easier.

- The Project window has been replaced with a **Project Navigation window**. Although it looks very similar, the new window is used only for navigation. Other functionality has been moved as described below.
- The **View button** that was in the upper left of the ribbon and many of the right-click functions on the objects in the Project window have been eliminated and moved to more intuitive locations.
- **All windows for an object are always displayed** when that object is active. The specific views shown will depend on the object type and how it is used. For example, when the main model is selected, it will always show tabs for Facility, Processes, Definitions, Data, Dashboard, and Results.
- The **Facility window** looks very similar, except that you will now see a Project Library listing all objects currently defined in your projects. Objects which are not place able in the current object (like the current object itself) are grayed out.
- The **Processes window** looks similar to the prior Process view, but without the tabs. The Tokens and Elements definitions have been moved to the new Definitions view.
- The **Definitions window** has been added (replacing the old Interface view) to unify into one location all of the things that *model builders* are most likely to define as they are building the model. Navigation within this view is similar to that within the Data view.
- The **Data window** is largely the same – containing the items that a *model user* is most likely to modify for a particular model run. Note that the creation of Lists has been moved to the Definitions view.
- The **Dashboard window** is largely unchanged, but now more readily available.
- The **Results window** is also largely unchanged except for consistency. The selection of reports and pivot grid is now on the left panel navigation.
- The **Project** now has a set of windows to add, edit, and remove its items (Models, Experiment, Symbols, Textures, and Path Decorators) as well as an Edit ribbon that allows creating/deleting models, subclassing from other objects, and protecting objects.

Other Features and UI Simplifications

- **Windows 7** is now supported.
- The combined **Project Navigation and Properties window** on the right now slides open or closed with a single click. Combining this with similar functionality on the library window, lets you dedicate almost a full screen to the facility window with just two clicks.
- Steps, elements and objects now have **type-on or F2 for rename in place**, without needing to go to the properties to rename it. In the facility window, we added "Rename" to rename objects in place to the context menu.
- Steps and Elements have **Basic and Advanced categories**. All categories except Basic are closed by default to avoid the distraction of options not commonly used.
- **Library Objects** will have all categories except logic closed by default to avoid the distraction of options not commonly used. Whenever any property has changed from its default value, that property will be in bold and its category will be open by default.
- **Auto-completer** has tool tips to help identify the options, responds to the up/down arrow and mouse wheel.
- **Nodes** and **Links** in the Facility Window now show a label with their name when they are selected
- Nodes have an **Appearance Tab** which allows you to add attached animation and in particular the drop down button on queues gives suggestions on which queues are available to animate.
- The attached queue and status pie now provide **Quick Create** items in their dropdowns, which automatically wire the drawn item to a specific simulation construct.
- The **trace window** is no longer cleared after encountering a runtime error -- the trace will persist to support debugging.
- Opening the **Dashboard** view of an object instance now brings it up in a floating, dockable window
- Added **ribbon buttons** for toggling the display of Status Objects, Buttons, and Links.
- **CurrentNodeID** function on Entity has been enhanced to return the NodeID (instead of 0) when the entity is in a station owned by the node, for example in parking.
- **Open Example** and **Open SimBit** are now available as sub-items of the Open item in the Application Menu
- If there is not enough space to show a full property value, we now pop-up a **tooltip** showing the value when you hover over it in the property grid.
- **Startup speed** has been improved and status dialogs added when a delay is unavoidable.
- **User-written Instance Descriptions** are now displayed when you hover over an object instance, step or element.
- **Subclass a library object** by right clicking on it. The new object will appear in the Navigation window and in Project Library.
- **Import Symbol** now directly supports image file types (jpeg, gif, bmp, png) to easily import images as rectangular symbol. This is particularly useful for importing floor plans and similar background items.
- Added new **Link functions** and results for including statistics on **NumberOfTravelors** and **TimeOnLink** and additional statistics in the results windows.
- Added additional **FrontTraffic functions** to allow easier decision logic based on what is ahead of an entity on a link.

Release 1.0, Sprint 28 – October 22, 2009

Resource States

Simio will now automatically track and report **TimeInState** statistics for the standard Resource, Server, Combiner and Separator objects. (e.g. **Time Failed, Time Starved, Time Idle**, etc).

- These statistics are recorded because of a new type of State that was added to Simio, called Lists States. These record statistics on the time that a state was a certain value. A user can also add a List State to their custom objects and set these to automatically change value based on the Resource state of the objects into which they are placed. This allows a user to easily record basic resource statistics, such as Time Busy, Time Idle, Time Failed, etc.
- The **Status Pie Charts** can now plot the values of these List States. Therefore, if Resource State is selected in the Status Pie, the chart will display the values of the Time Failed, Time Blocked, etc. of that object.

Dynamic Selection Rules

We have added new dynamic selection rules for doing campaigns (e.g. a series of related selections). For example **Campaign Up** would allow selection of the same or next higher size. This also includes **Campaign Down** and **Campaign Cycle**. These new dynamic selection rules can be found on the standard Resource object, Server object and WorkStation object. They can also be applied in the Routing Group element.

Set a Token or Object to Reference Multiple Tables

A token or agent can now be bound to multiple **data tables** using the SetTable step. Properties from two different tables can be referenced from the same entity and even within in a single expression.

New Location Property for Objects

A **Location property** was added to facility objects, which allow a user to explicitly specify an object's specific location in space (its height off the ground, for example).

- The Location property has an X value, Y value and Z value so the user can indicate exactly where this object should be placed within the Facility Window.

SimBits

- We have added a new SimBit (**SimpleTank**) demonstrating how to model and animate a tank filling and emptying by using a Level State.

General Improvements

- The modeling of entity congestion on paths has been optimized. Models that have very intensive entity accumulation on paths may run more than **50% faster**.
- We have **improved statistics** handling to make them clearer as well as fixing how some "corner cases" were handled in experiment processing.
- We have added a **new Step behavior**. When you are viewing the Process window, the default is to move to the next process step. When you are viewing other windows the default is now to move to the next entity transfer (a larger step size that is more intuitive when viewing the animation).
- The positions of **link vertices** (interim points between end points) may now be adjusted.
- **Pie charts** are improved by adding a legend and percentages on each wedge.
- When the facility window is resized, the **zooming** behaves more intuitively.

Simio Release 1.0, Sprint 27 – September 27, 2009

On-Screen Graphics and Statistics

We have added new display objects, enhanced existing objects, and allowed use of those objects in some novel ways. This provides dramatic new capabilities to communicate model results and behavior.

- Added a **Status Pie object** which allows you to show the values of a list of expressions (in percentages of their total summation).
- Allow the **Status Plot** to now plot more than one expression. The plot can now have a variety of styles applied to it as well, which change the background, label, and plot lines colors.
- **Floor Labels** can now intermix the value of expressions with text. To specify an expression put it between curly braces. For example, a label with the text "The current day is: {Math.Floor(Run.TimeNow/24)}", at hour 50, would show "The current day is: 2".
- Status Labels, Status Plots, Status Pies, Floor Labels, Circular Gauges, Linear Gauges, and Buttons can now be placed in the **Facility view**. They can also be attached to objects in the Facility view. If they are attached the Entities or Transporters, they will travel with those dynamic instances. They can be placed in the External and Dashboard views as well.
- Resource statistics on both '**UnitsAllocated'** and '**UnitsScheduled'** are now automatically displayed in the results view.

Arrival Times in Tables

We added features to make it easier to create **entity and material arrivals** from data in a table. When these tables are linked to and imported from an external file it provides powerful external initialization capabilities.

- A Timer element can now fire at times specified by a numeric property in rows in a table – see new **Arrival Table option** under Interval Type.
- A Source object can now generate entities based on arrivals in a table – see new **Arrival Table option** under Arrival Mode.
- A SimBit named **ScheduledMaterialArrivals** has been added to illustrate these concepts.

New Functions and Properties

We made significant changes to the object functions to provide more and better organized information and make the functions easier to use.

- New resource **allocation and capacity-related functions** added to every object. The new functions may be found under the 'Capacity' and 'SeizedObjects' name-spaces. Some existing functions were obsoleted by the new functions, and are now hidden from displays, but may still be used.
- New **capacity-related functions** added to every station element. The new functions may be found under the 'Capacity' name-spaces. Some existing functions were obsoleted by the new functions, and are now hidden from displays, but may still be used.
- Added **Math.Pow** (Number, Exponent) to enable easier math expressions involving exponents.
- Added **Math.If** (Condition, TrueResult, FalseResult) to easier and less obscure use of logical conditions in a numeric expression.
- Added a **Release Condition** property to Release step, to allow selective releasing of objects if multiple objects are owned.

Model Validation

We have improved Trace and Step to make it easier to understand the behavior of your model.

- **Trace can now be filtered** by Object, Entity, and Process. This allows you to hide parts of the trace that you are not interested in to make the interesting sections easier to analyze. To set a filter, select a trace line that contains something you want to filter on, then right click and select the desired filter item in the menu.
- **Stepping** in the process window is now by Step-instance rather than by calendar event. This provides a finer degree of control while analyzing model behavior.

Performance Optimization

- File load-times are now **10-20% faster**.

Decision Processes

We have improved the robustness of **decision processes** (processes invoked to make an immediate decision with no time advances) to eliminate potential misuse that could result in errors.

- We now restrict processes that are used as "decision processes" to containing only Assign, Decide, Search, and Execute steps.
- Since a decision process must run to completion without any interruption, we now ignore breakpoints, process suspension, and single-stepping when running a decision process.

Miscellaneous

- To eliminate potential errors, **processes** can no longer be edited while in run mode.
- Objects with "**Lock Edits**" (formerly known as "Lock Movements") cannot be either deleted or moved. This is particularly useful in preventing accidental movement or deletion of objects used as background or floor layout.
- Long tooltips for popup menus are now wrapped to make them more readable
- Attached objects are now "scoped" to the object to which they are attached to make reference easier.

Simio Release 1.0, Sprint 26 – September 7, 2009

Optimization of Execution Speed and Trace

- We have added Step Exclusion features to allow users to better optimize the performance of custom objects and we have used those same features to optimize Standard Library Objects. Depending on your model, you will find that our standard library executes from 20-50% faster compared to the previous sprint.
- We have also significantly reduced trace. Excluding steps has eliminated much of the "uninteresting" trace resulting in up to 60% less trace, with no loss of meaningful content. We have also improved the trace formatting. These changes will make it easier to understand and follow what is happening in the trace.

Arrays/Tables

- A state array's **size and initial values can now be "bound" to a data table.** If you chose the new [Table] option in the Dimension property of a state, it will bind that array to a table. When bound, the array size will be automatically initialized from the number of rows and numeric columns in the table, and the array data will be initialized from the data in the table as well.
- Look for the new **Import From** button on the Table ribbon. This allows import data from Excel or CSV into a table. (See Grid Data Providers below).
- Look for the new **Bind To** button on the Table ribbon. This allows you to bind a table to an external data source and cause it to be automatically refreshed from the external data just before a simulation runs.

Model Building Support

- On the project Home ribbon tab, you will find a new **Processor** option under the **New Model > Fixed Class** button. This takes the tedium out of creating a new object by automatically adding basic "plumbing". It creates a Fixed model with an input and output node, a station, external view, and a stub process for doing things. Everything is created and wired up, waiting for the addition of your process code to complete the object.
- On the project Home ribbon tab, you will find a **Select Add-In** button. This allows for user coded design add-Ins, which allows .NET code to do basic design in the Facility view. This is useful for doing things like translating data files into models, or quickly building ready-made models. A sample is included that creates a Source-Server-Sink model in one click. The code for this and other user code extensions can be found under \Examples\UserExtensions.
- Allow for user coded **Grid Data Providers**, which allows .NET code to provide "grid shaped" data for use in importing into Simio Data Tables. Two examples are included, one for reading data from CSV files, and one for reading data from Excel spreadsheets. Someone with programming background can add custom interfaces for any import source including your own custom database.

Step Exclusion Expression feature (Process View)

- You may now specify an Exclusion Expression for any step in a process. This feature is useful for temporarily removing logic from your model or optimizing your object's performance by only executing steps pertinent to property values entered by the user of the object.
- If specified, this expression will be evaluated at the start of a simulation run to determine if this step should be excluded from the run. If this expression evaluates to 1, this step will be excluded, and tokens will flow directly to its primary exit. If this expression evaluates to 2, and this step has a secondary exit, the step will be excluded, and tokens will flow directly to its

secondary exit. If the expression evaluates to any other value (typically 0), the step will remain active in the process.

Please refer to important compatibility notes in Known Anomalies (See Simio Start Page)

Miscellaneous Changes

- A **Dynamic Selection Rule** was added to the Station element. It allows you to specify a user-defined rule for selecting the next entity to enter the station. Server, Workstation, and Separator objects take advantage of this. If there is a zero-capacity input buffer and a dynamic selection rule is being applied, the next entity to be processed by the object will be dynamically selected from the upstream links/buffers connecting into the object.
- **User defined Selection Rules** are now available in the RoutingGroup
- Added time units for a **changeover matrix**
- Copying a process now copies the **image of the process** to the clipboard as well. This is particularly useful for creating external documentation of your processes.
- An Intelligent Object is the common base class of all objects that allows it to be seized and released as a resource. The **resource allocation queue** of an Intelligent Object is now a regular queue state, which means it can be queried and animated just like any other queue state.

Release 1.0, Sprint 25 – August 15, 2009

Arrays - Multi-dimensional array support has been added.
A Discrete State can be specified as an array or a matrix by changing the value of its Dimension property.

- If the Dimension property is set to **'Scalar'**, the State does not have a dimension and is therefore a scalar Discrete State.
- If the Dimension property is set to **'Vector'**, the State is 1 dimensional and the Rows property will determine the number of rows in the array.
- If the Dimension property is set to **'Matrix'**, the Rows and Columns properties will determine the number of rows and columns in the matrix.
- Up to **10 dimensions are supported** by setting Dimension to an integer from 0 to 10.

In order to reference a State Array, use the syntax "**StateArrayName[Index1, Index2, Index3, …]**".

- For example you could specify StateArray[2], where StateArray is the name of the State variable and 2 is the value of the row that is referenced.
- A multi dimensional State Array can be referenced similarly, such as StateArray[2,3] where 2,3 is the index to the matrix in this 2 dimensional array.
- State Array indexing is 1 based.
- For now initialization may be done by applying the initial value to the entire array or by using Read or Assigns. Additional initialization options will be made available in the next sprint.

Workstation - The Workstation object is now out of beta, ready for use, and appears in the Standard Library. Below is a quick overview of the features in Workstation:

- Workstation may be used to model a constrained resource that has a capacity of one. The Workstation may always be available (**fixed capacity**) or follow a **schedule**.
- Processing through the Workstation object is performed as a sequence of three activities: **setup**, **processing**, and **teardown**. The sequence of activities is referred to as an operation. The setup time can be Specific, Change-Dependent, or Sequence-Dependent. A Specific time can be a constant or be product/order dependent by referencing a table. Change/Sequence dependent times can be based on any characteristic of the product (e.g. size or color).
- The entity that is processed through the workstation represents a **production lot**. The data for a production lot is typically held in data tables, and then referenced by the entity.
- The **operation quantity** specifies the number of "items" in the production lot that is represented by each arriving entity. This value can be a constant or be product/order dependent by referencing a table. The lot can be processed as a single production unit or can be broken into a sequence of **batches** that are processed one at a time until the entire lot has been produced.
- **Material** or a **Bill of material** may be required to start the processing of each batch. Material or a bill of material may also be produced at the end of each batch processing. A batch of produced material may have a transfer time before it becomes available for consumption by other workstations to easily model situations involving intermediate transfer batching.
- The workstation object is considered to be a primary resource that is required during the entire operation: i.e. setup, the processing of all batches, and teardown. In addition, any number of **secondary resources** may be required and held by each batch across one or more of the three activities. Secondary resources can be specific resources or selected based on rules from a list of resources.

- A workstation has an input and output buffer. Either may be zero and contribute to **blocking** between workstations. The next production lot to be processed can be selected based on both a **static ranking** and a **dynamic selection process**.
- A production lot may have a **maximum make span** that limits when it can be started based on the availability of the primary resource. The maximum make span may be a constant or order/product dependent by referencing a table. The make span check verifies that that the primary resource is available to complete the operation within the maximum make span.
- A workstation has **failures** based on calendar time, processing count, or event count.
- A workstation has **add-on process triggers** that allow a user to embed custom logic inside the workstation (e.g. seize a worker prior to a repair). These add on processes can be used to alter or extend the standard behavior of the workstation.

Process Window - We made a series of changes to make the process window easier to use:

- All processes are now sorted alphabetically.
- A process can now be under a category. Categories are sorted alphabetically as well.
- Process categories are collapsible to allow hiding categories not of current interest.
- You can "zoom" an individual process to make it easier to read by clicking on it and using the "+" and "-" keys on your keyboard.
- The display of Step instance names are now longer to improve process self-documentation
- Several other UI improvements and bug fixes

SimBits

- SimBits are now documented in a consistent format to improve readability.
- SimBits are now described in the help. They can be found categorized on the main SimBit page and listed alphabetically within the SimBit area.
- SimBit content is searchable, so now if you search a term (like Vehicle), you will not only find all the help topics that discuss the term, but also all the SimBits that feature that term.
- We added nine new SimBits including examples of vehicle control, object protection, writing to a file, and moveable resources.

Miscellaneous changes:

- The "Composite" tab of the Schedule Data view now shows the **composite schedule** (repeating Cycle with Exceptions laid over it) for a schedule.
- The **alignment of objects in a queue** can now be changed to follow the direction of the queue as well as just always pointing to the left (as they do now).
- By popular demand, we **changed the interface for drawing a line** to make it more intuitive. Right click no longer "commits" the point that you right-clicked for line, polyline, queue, polygon, and closed curve. Instead, right click ends the line at the last point previously clicked.
- Provided additional model information for user-defined steps and elements.
- Added support for **mapping repeating properties** to other repeating properties for more flexible object building.
- Added support for editing **enumeration captions** and **list member captions** for more flexible object building.
- In property definitions **category name fields** can now be selected from a list of existing category names.

User Interface changes:

- **Floor labels** provide a rectangular section of text drawn on the floor. The text can have simple formatting such as italic, bold, underline, and color. This is useful for labeling and commenting your model.
- **Status Labels** can now be attached to a dynamic object (like an entity), in the same manner that Attached Queues can. To use them, simply select the object, click on the "Attached Status Label" button in the "Symbols" tab of the "Object Tools" category, and draw the label. The label's expression should involve the instance which you are attaching it to. For example a label with "ModelEntity1.MyState" on an instance of ModelEntity called "ModelEntity1" would animate the custom state value "MyState" on every dynamic object coming from "ModelEntity1".
- **Named views** now allow you to place the camera at a location, and save that view for later. You can either pick a specific view to go to, or cycle through them. See the "Named Views" group of the "View" tab of the "Facility Tools" category.
- Significant improvements were made in **execution speed** (much more coming).
- Now the **watch window includes Processes** (and their active tokens) under their parent object.
- Added "**state functions**", so at runtime you can ask for information about a state. Right now, they are only on Queue States. For example, you could make decisions based on the average number waiting in the entry queue of a station by using an expression like "Station1.EntryQueue.AverageNumberWaiting < 5"
- Automatically bring up the **trace** window when opening a project if trace was turned on. This helps avoid running with trace accidentally turned on.
- **Enhanced DXF import** to allow more files to load, and improve error reporting for problems.
- **Categories** specified on property definitions can now be selected from a pull-down list.
- Improved the **software version** stamp on the Start page for clarity.

Modeling Engine changes:

- **Search** step can now search queues, and you can save the found object count and found object index from the search.
- **Route** step now has SelectionCondition to optionally filter the list of candidate destinations.
- **Seize** step now has a SelectionCondition to optionally filter the list of candidates to seize.
- **BasicNode** and **TransferNode** were optimized some (more coming) to execute fewer steps per entity per node. This improves model execution speed as well as reduces the amount of trace.
- **Storage** element was added to define a custom queue state, with **Insert** and **Remove** steps to insert and remove objects into the storage element's queue.
- New **CanEnterObjects** property on entity objects allows you to control whether entities of an instance type can enter other objects.
- New **OnEnteredFromAssociatedObject** and **OnEnteredToAssociatedObject** built-in node processes that may be used to optimize the design of node process logic. If these processes are not used or applicable, then the standard **OnEntered** process is executed by default for the node.
- Added "**Is.*ClassName***", "**Is.*InstanceName***", "**Candidate.Is.*ClassName***", and "**Candidate.Is.*InstanceName***". This allows easier logic based on object characteristics. For example to select a path based on entity instance, you could specify a path weight of Is.PartA on one path and Is.PartB on another path. This function also works with base classes like "Is.Node".
- Added user-requested rate unit: **Meters per Minute**.
    - _____

Added a **Watch Window** to allow you to watch object details during a run. You can "drill into" each object when the run is paused to determine the current values of its states, functions, and child elements, including any queues and entities in those queues.

Added support for six new distributions, bringing the total to twenty**:**

- **PearsonVI** (ShapeA, ShapeB, Scale) - This distribution has a range from 0 to infinity and is typically used to model a task time.
- **LogLogistic** (Shape, Scale) - This distribution has a range from 0 to infinity and is typically used to model a task time.
- **Poisson** (Mean) - The number of occurrences in an interval of time when the events are occurring at a constant rate according to a Poisson process.
- **Binomial** (ProbababilityOfSuccess, NumberTrails) - Number of successes in a specified number of independent trails.
- **Geometric** (ProbabilityOfSuccess) - The number of failures before the first success. The parameter for distribution specifies the probability of success for each independent trial.
- **NegativeBinomial** (ProbabilityOfSuccess, NumberOfSuccesses) - The number of failures before reaching the specified number of successes.

Improved the Dashboard get a dynamic view of the object's state over time.

- Added **circular gauges** with eight style choices.
- Added **linear gauges** with eight style choices.
- Enhanced **status plot** with new look and options.

Added **DXF Import** capabilities. Import Symbol now supports the file type of *AutoCAD Drawing Exchange Format* (dxf). Objects saved in this popular format can now be directly imported into Simio.

Added Subscribe and UnSubscribe steps. The **Subscribe** step may be used to add a new triggering event for a process, indicating that the process is to be executed if the event occurs. Use the **UnSubscribe** step to cancel a triggering event for a process. Though simple, these two steps add some powerful application abilities, as an object's process logic can now dynamically add and cancel triggering events for a process. You are no longer limited to a single triggering event that has to be specified at design time only.

Miscellaneous changes:

- Added four new **SimBits** and a new **Regional Health Care example**.
- Added support for **command line-based authentication** to make automated deployment of Simio easier.
- Replaced the algorithm used in calculating **normal distribution** to a more robust and slightly more accurate algorithm.
- Added an **experiment trace window** for displaying messages about runs executing in experiments.
- The **Workstation** object has been improved, but it is still beta and not ready for use.
- Fixed a problem specifying response expressions in experiments.
- Server and Separator now support **Dynamic Selection** from their input buffer.
- Added lots more **help content** and started reformatting for improved consistency.
- Improved execution speed for running experiments.
- Disallowed certain property changes during runtime that sometimes caused model instability.

- Added capability for academic users to read projects that were saved in Beta software.

Simio Release 1.0, Sprint 22 – June 14, 2009

We are focusing primarily on stabilizing the product to ensure that all existing features work as designed and we are eliminating nagging little problems that make modeling difficult.

Some specific enhancements:

- **Activation**: A hyperlink now appears at the top right corner of the application window when in evaluation mode to allow you to enter your license key. And the license authentication dialog can now email authentication requests directly to us.
- Added a warning when overall number of dynamic objects in the system hits 20,000 and clarified the related message that displays when you exceed the designated number of entities of a particular type. Both of these **help you identify modeling errors** that cause run-away entity counts.
- **Queues**: Added "View Queues" button for turning display of queues on and off. Changed color for selected queues to turn them darker when selected
- We now highlight a **node** when you are in link drawing mode to indicate that if you click the link will be anchored there
- We now use the **project file name** as the project name, unless the user explicitly provides a different one
- **Symbols** and **external views** now have their center point determined by the center of the bounding box of all objects in the view, rather than the (0,0) point in the symbol drawing window.
- You can now reference **Entity classes** defined in your project but not yet instantiated in your model in properties and expressions.
- Changing the default value for a **property** now affects the model if that property is referenced by objects in the model.
- Can now use a property's DisplayName as column heading in **table view**.
- **Animation objects** are now able to show errors in the error window.
- In those cases when you encounter an **Exception** we now provide the opportunity to attach your model along with the exception report. The first time you receive a specific exception, please send it (and the model) to us. This is very helpful in eliminating future problems of this type.
- Better support **Foreign references** (e.g. references normally outside the scope of the containing object). For additional detail search for the "Candidate Expressions" topic in Simio help.
- Added **more SimBits** and **on-line Training Videos** (see Simio Start Page).
- Added support for temporary (time-expiring) activations.
- Added better conversion of old models. If you have had trouble running a model developed in pre-release software, please try again in this build. (Be sure to answer Yes to the question on using new library objects). Many more models will now come through working. If you have a large model that you are still having trouble with, please send it to us and we *may* be able to help.
- We added support for three new distributions: **Pert**, **JohnsonSB**, and **JohnsonSU**.

Simio Release 1.0, Sprint 21 – May 22, 2009

This is the initial public release of Simio version 1.0. This space will contain a list of what has changed since the initial release. Since that list is currently empty (this is the initial release), we thought we might take a minute and describe what you are getting – it is not the typical "release 1.0" product.

As you might guess from the sprint number, this is actually the 21st release that has been distributed (http://www.simio.com/downloads/public/documents/SimioHistoricalReleaseNotes.pdf) over the last 16 months. Each individual build (so far well over 3000) must pass a set of nearly 2000 automatic tests before it reaches any human contact. After that it must pass our internal testing staff for whom beating up software is a passion. Then, at the end of each three week sprint it gets posted for others to use.

We currently have several hundred Simio Insiders, Beta testers, and evaluators who have been putting each sprint through its paces and giving us great feedback on any issues, missing features, or confusing user interfaces that they find. As a result of this extensive testing and user involvement, we have a stable, high quality product. Our team is very proud of what we have accomplished in terms of both technological breakthroughs and quality and we hope that you are pleased with our work.

Yes, Simio is still an early product and it's not perfect, but we are not stopping to rest (OK, everyone took off Monday because it's a US holiday ☺). We will continue our agile development and we will have new software available every three weeks. So if you find a way we can improve, tell us. You might be surprised at how fast a good idea or a fix to a problem can show up in new software.

We all wish you the best experience possible.
The Simio Team
(Dennis, DaveS, Rich, DaveT, Cory, Glenn, Judy, Renee, Christine, Samantha, and Tom)

Simio Sprint 20B – May 15, 2009

No new software features in this interim build. We are focusing completely on stabilizing the product to ensure that all existing features work as designed. So lots of things work much better than they previously did. But do look for our on-line help by clicking the '?' in the upper right. While still not complete, you will find quite a bit of useful information there.

Simio Sprint 20 – May 6, 2009

We have a new **on-line support system** to make getting technical support and submitting suggestions even easier. This support site also has an on-line knowledgebase for instantly providing solutions to common problems. You can find the link on the Simio Start Page. When you first go there and register you will get immediate access to the searchable knowledgebase to find answers to common issues. After your registration is approved, you will also be able to submit support and enhancement requests interactively and track their progress at any time.
While you are looking at the **Start Page**, notice the new layout and information. **Examples** and **SimBits** are now accessible via the Start Page. SimBits are small models with accompanying pdf files to describe how to model a specific situation. When you load a SimBit it will also load its companion pdf file to explain the modeling techniques illustrated. You will see lots more SimBits to come – your topic suggestions are welcome.
We completely replaced our old symbol library with **over one hundred new symbols** across nine categories. These new symbols are smaller "low poly" symbols which make it easier to build quick

models while keeping model size small. Of course you can still easily import from Google Warehouse. Now Sketchup 7 symbols are supported too.
We have added three **new tabular reports** to supplement the Pivot Tables. The **Interactive report** provides the Value of each statistic from an interactive run. Since it has only a single replication and parameters may have been changed while running, this report should generally not be relied on for statistical results. Reports run from the Experiment Window will generally have multiple replications and instead of reporting the Value of each statistic, will report the Average, Half-Width, Minimum, and Maximum across all replications run in that scenario. The **Scenario Detail report** summarizes all the data from each specific scenario run in the experiment. The **Scenario Comparison report** makes it easy to compare how each statistic performed across all the scenarios run.
In the **Experiment Window** and **Interactive Results**, you can now export report results to a CSV file. You can also directly change Run Setup parameters from the Experiment ribbon. We also implemented miscellaneous cleanup and bug fixes.
Schedules now work much more reliably. You can create simple or complex schedules from the Data Window, and then reference them from Standard Library objects like Server. We also now support capacity-based schedules to allow the resource capacity to be varied between different values.

Getting started and exploring the product are a bit easier. We added labels that show up when views are empty of contents, to indicate how to start adding things in those views. We have also added extensive tool tips to many more items including steps, elements, and library objects – when you hover the mouse over them, you will see tips on their use.
We **simplified object interaction**. You can no longer open the Facility or External view for a Link or Node, since they are not supported. Double clicking on a Fixed object in the Project Contents opens its Facility view, double clicking on any other type of object now opens its Process view.
Path decorators are improved. We recreated our existing path decorators to use real-world measurements. We also added a "Single Lane" road path decorator.
We added the option to **select if a texture should be tiled** when applied. Turning off tiling allows you to place larger images (jpeg files) on your model. We also added a related capability to **lock movement** (via right click). Both of these apply to Rectangles, Polygons, Ellipses and Closed Curves. This combination allows you to make an image become the "floor" (e.g. background) without it interfering with model navigation. See updated **MiningExample** for an illustration.
We have improved the **Workstation Object** however it is still incomplete and only lightly tested. So we also created a **Beta Library** that can be found parallel to the Standard Library and we moved Workstation to this new location so that it would not accidentally be used in a project.
You can now pass a **StateProperty** into an object and use it both in fields looking for a state property (like Monitor or Statistics elements) as well as in an expression that is looking for the value of a State.
[Compatibility alert] We **changed syntax** for referencing your parent object to eliminate some areas where our old notation was ambiguous. When you were inside on object you used to have to refer to its properties by preceding the property name with the object definition name. This could have caused ambiguity if you also had an associated object of the same definition. Now to reference properties associated with the object itself you may reference the property directly or use the preface "My". For example when you are in the object Model1 and want to reference its property named Capacity; instead of referencing it as "Model.Capacity" you would reference "Capacity" or "My.Capacity".
[Compatibility alert] We have enhanced **identifier validation**. Spaces and special characters are no longer allowed in identifiers (identifiers can contain letters, digits, and underscores, but cannot start with a digit). Identical identifiers are no longer permitted within the same context. Automatic naming is now done more consistently.
[Compatibility alert] We have significantly enhanced the **FindObject** step and renamed it to the **Search** step. You can now search owned object lists (e.g. resource objects that an object has seized) as well as specifying parameters like search direction, starting index, match condition, limit, and return value.

Lots of **bug fixes** and **minor enhancements** including:

- Our executable is now called Simio.exe instead of SimioDesign.exe
- Several internal changes for stabilizations and reduced memory usage for graphics.
- Expression properties for animation now use our expression auto-completer.
- We now animate entities in free space.
- Miscellaneous fixes in table window.
- Added public/private support to states and elements.
- Enhanced Pickup and Dropoff and Vehicle Loaded and Unloaded add-on processes.
- The Produce and Consume steps were extended to work with activities
- The 3d renderer will now fall back to software rendering if no hardware is found.
- Enhanced Station element to allow capacity to be assigned during a run.
- Enhanced RoutingLogic element by adding a BlockedCondition (allowing users to define the meaning of “Blocked”).

Simio Sprint 19 – April 10, 2009

We are winding down on release 1.0. Our primary emphasis now is on stabilization and performance, although we are still adding some improvements along the way. Here are some highlights of our progress:

We have improved **batch execution speed** and memory usage.

We have **improved startup**. The Simio user interface now appears much faster at startup. We are now loading things like Symbols in the background, which you can see if you drop down the "Place Symbol" gallery and see symbols appearing in it as you look at it.

We added **Cut, Copy, and Paste** support for steps, processes, and all simulation objects except links. We do not yet have multi-select, but this provides significant new functionality.

We have added a **Most Recent File List** to the application button and prepopulated it with some example files to give you a quick start. Files can be optionally "pinned" to this list as the examples are, to make them stay visible until they are unpinned.

We have improved the user interface of the **Schedule editor** and added use of Work Schedules to many standard library objects.

To **simplify use of the Table view**, we removed "design mode" and removed the section of the window that held the list of properties in the table. Now, clicking on the column heading in the remaining data entry area now selects the defining property for that column.

At long last we fixed the bugs in the auto-completer (**expression editor**) so it now works correctly when a mouse is used.

We removed the pre-loaded "Name" column from **Data Tables and Sequence Tables** to simplify their use. Name is not required by this version of Simio.

We have **added new keyboard shortcuts** that are commonly used like F5 to run, F6 to step, Ctrl-C to copy, Ctrl-V to paste. We have also significantly improved the quantity and comprehensiveness of our tool tips to make it easier to learn Simio.

By popular demand, we **added a "Picture" state** to the ModelEntity that gets created for a user by default. The default "Current Symbol Index" expressions have also been updated with default expressions to make animating changes in object state easier. Now entities use the "Picture" state to get the index of the symbol to show, all others use an expression indicating if they are currently seized or not. It is now much easier to use the Add Additional Symbol capability.

Overriding a process now copies the contents of the old process into the new process, making it easier to make small changes to an existing process.

We fixed dozens of big and small problems that have been reported by users, so the software should work noticeably better – **keep those reports coming**! We also implemented better error trapping and messaging so that when something goes wrong, it is easier for you to diagnose.

And finally, while our goal is to eliminate the unhelpful "Internal Error" runtime message, we have added a better interface for it that gives you the option of sending us an email to help us find new problems. **Please help us by selecting the option to email the "Internal Error" problem.** Doing so will make it less likely that you will see that error in the future. By the way, many people did not realize that it is generally safe to continue after such an error. It is not generally necessary to shut down Simio after an Internal Error. Just "Continue" and you will not lose your work.

Simio Sprint 18 – March 22, 2009

This sprint includes a lot of User Interface improvements and enhancements. As we draw closer to our formal 1.0 release of Simio, it is exciting to see how much has been accomplished in these past three weeks.

Windows and Views

We have improved the Trace window:

- It now displays without "flashing" while running the trace, and displays which file it is tracing to in the title bar.
- Trace messages are no longer wiped clean in the Trace window from run start-up.
- Miscellaneous trace and error message improvements to make them more useful and easier to read.

We have added a new Project Graphics View window, which allows you to see of the graphical things that have been used in your project (Symbols, Textures, and Path Decorators). From this view you can right click on an item and choose to edit it or remove it from the project.

In drawing windows, we've renamed "Top" and "Perspective" view orientations to be "2D" and "3D". The shortcut keys are now "2" and "3" instead of "T" and "P".

We have removed the "Project Library" from the left side, and made it possible to drag and drop objects from your Project Contents into a model. For example, to create an instance of "ModelEntity" inside of "Model", now you would open the facility view of "Model", and drag the "ModelEntity" from the Project Contents into the facility.

The Schedule window has been simplified so now there is just one kind of schedule, that has both a recurring part, and a list of exceptions.

There is rudimentary support for importing and exporting to and from a CSV file in the Table view.

Ribbon menus

There is a new button called the Add Texture button to add a texture to the project, which allows you to pull in an image file to use as a texture straight into your project. Before, you would have had to copy the image to the "Skins" directory under the Simio directory in order to have it show up in your Texture library. These changes are accompanied by other ribbon updates and overall user interface adjustments. For example, button resizing, icon replacements and minor terminology changes.

Many tooltips were added for buttons on the ribbon. These provide additional information and help about how things are used, and how things fit together.

We have updated the animation speed adjustments. Now we have a "Time Between", which is the simulation time to execute between animation updates (expressed in tenths of a second), and the "Adjust Speed" slider, which now adjusts the last entered value in "Time Between", +/- 50%. If you need to go much faster (or much slower), now you should simply enter a new value in "Time Between". A larger value will make the animation less frequent and thus faster, but perhaps 'choppy'; a lower value will make the animation updates more frequent, and thus smoother, but slower.

Properties

The drop-down lists for certain property types now include properties from tables. And the auto-completer's drop-down list now includes numeric properties from tables in the model and entity names from the project.

Property names are no longer case sensitive. WARNING – In a future build we will probably further restrict all names. To assure model compatibility do not use spaces and special characters other than '_' in any names (property names, object names, …).

We have added Event and String properties to the list of properties you can create.

You can now specify the color of a Node in its definition. If you create a Node type model, in its properties you will see "Node Color", changing this will affect the color of the node when it is

instantiated in a model. The color of the Basic Node and Transfer Node in the Standard Library were changed as well.

Security

We have implemented model protection. To protect a model, right click on it in the Project Contents window and select Protection->Protect Model…, this will prompt for a password. Once the model is protected, a user will not be able to edit or subclass the model without providing the password.

Object Behavior

We have improved "Sub-classing" objects. By default, dragging and dropping an object from the library on the left to your Project Contents on the right, now subclasses the object instead of copies it. By holding down the Ctrl key you can change the drop operation from subclass to copy, if you need to do that.

The *Destroyable* property on Agent, Entity, or Transporter instance allows you to specify whether a dynamic object can be destroyed using the Destroy step.

Steps

The **FindObject** step was enhanced to allow you to search any object instance or object list to get a token pointer to an object.

The **EnterStation** and **ExitStation** steps were eliminated and their functionality merged into the **Transfer** step.

The **Transfer** step now allows you to do additional types of entity transfers, such as from free space to a node, from a node to another node, or from a node to free space.

The **Batch** step behavior changed so that batch quantities are only determined by the parent entities.

Reports

Reports can now include half-width as well as model name, project name, and author.

Experiments

Experimentation now calculates half-widths across replications.

Several issues were fixed that caused incorrect statistics to be displayed.

Simio Sprint 17 – February 27, 2009

This is another pretty full release. Changes were made across the entire product. This is our first Beta release and is pretty close to feature complete.

Standard Library

We are no longer displaying the "DefaultEntity" until you want it. You only need to display it if you want to change its default picture or properties. When you manually place the default entity (or any other) it will be displayed.

Object Interfaces:

- **BasicNode** has new 'Entered' and 'Exited' events.
- **TransferNode** has new 'Entered', 'Exited', and 'RiderWaiting' events.
- **Vehicle** has new 'EnteredNode' add-on process trigger. New 'Pickup' and 'Dropoff' public processes to force pickup or dropoff attempts from logic (these are essentially 'methods' that you can run on the object via an **Execute** step).
- **Path, TimePath, Connector, Conveyor** have new 'Close' and 'Open' public processes that allow you to easily close/open entry onto the link via logic.

Adjusted how the **Link size** works. Size.Length now reports the size used by the simulation engine for calculations (either the drawn size or the logical size depending on the value of "Drawn to Scale"). Size.Width is now used by any path decorators applied to the line. Adjusting it will scale them appropriately.

Added functionality to "**Transfer from FreeSpace** to Node" in the Transfer step.

'Departed' terminology has been changed to 'Exited' for more intuitive and flexible reference.

Drawing Ribbon

The **height, line width,** and **line style** for drawing objects was moved to controls on the ribbon.

Texture mapping is now more consistent between objects. All solid objects use the same 'scale' when texturing themselves. Skins can now be applied separately to the tops and sides of objects (rectangle, ellipse, polygon, closed curve)

Added "**Clear Material**" button, to remove any material application from an object.

The last thing selected is now shown in the Place Symbol, Color, and Skin button. Clicking on the top part of the button allows you to place or apply that same thing again.

Process View

A new **Tokens tab** allows you to create your own types of Tokens for use in a process. These are nice if you want to do things like counts or other calculations that require local variable storage, but don't necessarily need the value to stick around in an object level state. A Token can contain as many states as you want.

User Interface

- We now show a **breakpoint indicator on a step** if it is set.
- Several UI improvements were made including better identification of End points and their movement, and highlighting the selected connector line when moving it.
- We simplified the UI by moving the Process selection list from the Process Window to the ribbon.

Steps and Elements

- **Arrive** has been renamed to **EndTransfer**.
- **DepartParent, DepartNode,** and **DepartLink** steps have been consolidated into a single **Transfer** step.
- **Output** element renamed to **OutputStatistic**
- Some unused Steps and Elements have been dropped for simplification.

3D View

Implemented object stacking. If you hold down the 'Ctrl' key while dragging an object, it will automatically stack on top of any other object it intersects.

Tables

Tables are a feature which allows you to define a flexible table of values and then easily reference it by a token, entity, or object.

Added a new **SetTable step** to assign a table and an active row to a token, entity, or parent object. (Note: SetSequence has been deprecated).

Added a new referencing scheme for properties in a table: a user can reference a property in the active row with **TableName.PropertyName**, or any row specified by an index with the syntax TableName[index].PropertyName.

Added a new **RandomRow** selection feature for representing product mix. RandomRow returns a random row index based on weights specified by a numeric table property. The random value is referenced by TableName.PropertyName.RandomRow, where PropertyName is the name of a numeric property in the table that specifies the weights. This is typically used for specifying the active row in the SetTable step and is very useful for modeling a data column that holds the product mix as a percent or fraction.

Added "Design Mode" to Table Window.

Miscellaneous

Combined Run and Pause buttons into a single button, so you can now alternate between them without having to re-aim the mouse.

Added a "View Rotate" button to the view tab.

Merged the Model List and Experiment List into a single Project View. Now Experiments are listed under their associated Model.

Added Breakpoint item to facility object's right-click menu and we now show an indicator on the object if a breakpoint is set.
Improved auto-complete within the **expression editor**, including look, tab key behavior, pop-up location.
Improved flexibility of object building tools. You can now make certain changes to inherited properties, including changing its DisplayName, Description, Category, and DefaultValue. You can also set if an inherited property is visible at all.
Can now select from multiple report files for both interactive and experiment results. The selected report is retained between use. Note that included report is currently just a simple sample. Formal reports are coming soon.
Added additional element-specific properties to the list of properties you can add.
Simplified user-definable states to Discrete and Continuous.
Status bar in Experiment view now reflects accurate run state.
The delete key, as well as using a right-click context menu allows you to delete things in the element, interface, changeover matrix, function table, list, and rate table views.

Simio Sprint 16 – February 6, 2009

This release is packed full of enhancements. Some enhancements were made to improve the beginning modeler's experience, such as improving the way new models are created. Other features we have added are recommended for more advanced model builders to try out. An example of this is the ability to define your own steps and elements.

User Interface

When you run Simio, the first thing you may notice is an "Evaluation" dialog asking for an **activation code**. You may press the "Continue Evaluation" button each time and continue using the software for 30 days. Or you can find the required code in the forum software download posting and activate it to remove the message.

Run Length has now been added as an option on the Run ribbon. This allows you to specify an absolute run time in addition to setting a calendar ending time.

We have added support for analyzing **Steady State** and **Terminating** runs and **Warm-up Time** for the former; look for new properties on the Experiment Window.

When saving a model, we now also **save the last view**. So if you are in 3D perspective view at some location, when you open the model again, you will have that same view.
Auto-complete content within the expression editor has been reorganized and improved. Much of the "clutter" has been removed. Properties and states of the top-level model are now found underneath the model's name. The current simulation time can now be retrieved from the TimeNow function on any object. To enter the expression editor, click on the down arrow at the right side of an expression box. Within the expression editor, type any character to bring up a list of all available functions/variables. Use your cursor (not yet the mouse) to select an option, then press "." to select a variable or move to the next part.

Documentation

We are in the process of coalescing our information into a single source. You can now access the **Getting Started Guide** within the help pages (see link on help welcome page). This information will gradually be migrated into the individual help topics.

We have added **Steps documentation** and **Elements documentation**. Look in the contents under Handy References -> Views -> The Process View.

We have added a help section on Simio Extensions for advanced users who want to create their own blocks and elements (see discussion below).

We are starting our effort to provide more complete formal documentation. **Please use the links at the bottom of each help page to provide feedback and suggestions.**

Library Objects

The **Resource** object in the standard library was significantly enhanced.

- **Reliability** options for resource downtimes were added.
- Several **add-On process triggers** were added. In particular, an 'Evaluating Allocation' trigger was added to provide external control over when/if a resource accepts an allocation attempt. This

can be very useful for modeling "smart" resource-centric models where the resource is making decisions on what to work on.
- An **Allocate** public process on the Resource that allows external logic to manually trigger an allocation attempt of the resource object's capacity.

The **Vehicle** object in the standard library was enhanced.
- **Evaluating Rider** add-on process trigger that provides external control over when an 'On Demand' Vehicle accepts a rider reservation request.
- **PlanPickup** public process that allows external logic to manually trigger a pickup reservation search by the vehicle in the system.
- New **Idle Action** choices of 'Go To Home' and 'None' were added to allow the vehicle to stop on a link in a potentially blocking state, rather than always move off the link into a parking area.

We added new options to the **outbound link selection** on **Nodes**. In addition to selecting the link based on Shortest Path or Link Weight, you can also further clarify it by selecting an Outbound Link Preference of Available, which will prevent selecting an unavailable link (e.g. it is busy or going the wrong direction) when another link is available.

Added **Time Varying Arrival Rate** to the Source Object to allow arrivals whose rate follows a Rate Pattern that varies over time. Also added **Rate Tables** as a new View under Data in which to specify the arrival rates for each time period.

We changed the default workflow for working with entities in your model, removing the "BasicEntity" from the Standard Library. Now when you click "New Model", you get two things added to your project, an **ModelEntity** object, and the model object. In the model object an instance of the entity object called **DefaultEntity** is automatically created and marked as the "default" entity for the model. When you drag in something that needs a reference to an entity instance (such as a Source), it will automatically pick up this instance as its value. Among other things, this makes it much easier to add a state to the entity to track something in the system. Now you can just go over to the model list of your project, right click on entity object, and choose the "Add State" menu, from which you can add a new state. Since the entity instance in your model is an instance of the entity object in your project, it automatically gets the new state.

Model Validation and Results

Added the ability to set **breakpoints** and for the simulation to stop when it hits a breakpoint. In the Facility View, breakpoints can be set on Fixed objects (which includes Nodes and Links), and will pause the simulation when an Entity is about to enter the Fixed object. In the Process View, breakpoints can be set on individual steps, and will pause the simulation when a Token is about to execute the step.
Improved **trace** content. We added new messages and content. We improved the identifiers used for entities and transporters – **dynamic objects** (entities) are identified with their object ID number in the form InstanceName.nnn; **static objects** (transporters) are identified with their sequential instance number in the form InstanceName[nnn]. We now present the identifying information in a way to more clearly identify a series of sequential activities – a row of blank identifiers indicates it remains identical to the previous line.
The **Pivot Table** in the Results View has been improved. We have added new statistics (like a throughput count on Links), relabeled some statistics for clarity, and suppressed some redundant statistics.

Object Creation

We implemented more functionality when **inheriting from another object**. You now inherit function tables, lists, schedules, changeover matrices, transfer points (in the external view), and most

importantly the facility view. When inheriting from another object, you will inherit all the logic, but none of the graphics. That means you will not inherit any graphics you put in your external, facility, or dashboard views. When inheriting a facility, you get the objects with their “default” symbols.

Process View

We **enhanced the process editing**. Now we label to “Begin”, “End”, and “Available” nodes. When you hover over an “Available” node we give you an indication you can click and drag it. We also now label the outbound connections of steps with two or more of them. For example, the Decide step connections are now labeled “True” and “False”. Also, we removed the requirement that all “Available” nodes must be connected to “End”, now if a token reaches an “Available” node, it simply acts as if went to “End”.

The set of **Common Steps** has been reduced to the nine steps that are most commonly used. The rest of the steps are still available in the All (A-Z) category. In particular, this should make it easier to define typical add-on processes.

A new **FindObject** step allows you to search a dynamic group of agents/entities/transporters and find an object meeting a Match Condition.

A new **Allocate** step allows you to manually trigger a reallocation attempt of a resource object’s capacity.

The internal behavior of the **EnterStation** step was improved.

The **Route** step behavior was enhanced related to the use of the BlockedRoutingRule feature, which allows the entity to choose only a non-blocked route from a list of possible destinations.

Advanced Features

You can now **extend Simio with your own Steps and Elements**. Create a new step with String, Numeric, Expression, State Reference, Element Reference, or Repeat Group properties and write them in any .NET language (i.e. C# or VB.NET). You can create a new Element with all the same types of properties as well as States and Events. You can schedule events on the calendar and can implement forms of blocking.

We have provided two **examples** in the Examples\UserExtensions directory. They are Visual Studio 2008 projects, so you will need a product like Visual Studio 2008 Express Edition to compile them. One example is a very simple text file read write. It implements a File element, and Read and Write steps. The other example is a BinaryGate, which provides a BinaryGate element, with Open, Close, and PassThru steps. The behavior of the gate is for tokens to queue up while the gate is closed, then to all be released at the same time when the gate is open. This is an example of custom blocking behavior.

You can also look in the Help, under the section “SimioExtensions Namespace” for information on the interfaces available.

Both of the UserExtension examples have been pre-compiled and put into the UserExtensions directory, so you will actually see them in the user interface and can use them in your models. In the process view you can open the “User Defined” steps pane to see the steps, and in the elements view there is a new group in the ribbon for User Defined elements, where you should see the “File” and “BinaryGate” elements.

Sprint 15 –January 16, 2009

Our primary focus in this release was to allow models to survive from one sprint to the next. A second emphasis was to revamp our previous auto-complete and expressions features. And finally, we improved the general robustness of the modeling features.

Files

We have **improved model compatibility** by updating object definitions to newer versions. So, if you have used an object in your model, and then load a library with a newer version of that object (one that has new properties or new processing logic), Simio will ask if you wish to update the objects in your model to the newer version. This also works if you have created a model in your project which is a subclass of another. If updates are made to the base definition, those updates will properly show up in your subclass (if you choose the option to update). Currently, we are updating properties, states, events, elements, and processes. We cannot yet guarantee that all models and objects will move forward, but the odds just got a lot better. The set of examples written early in Sprint 13 are still surviving - hopefully your models will survive too.

We now show **loading and saving informational dialogs** when loading and saving a project. The loading dialog displays the relative progress of the load.

We have implemented **file compression** into our project files so that they take about 1/3 to 1/2 the space they used to.

Expressions

Redesigned **expression editing** and auto-complete feature. It now shows up in a "fly-out" window that is available when editing an expression property in the property grid. Click the arrow at the end of an *expression* field to get a wide data entry field. When you type anything into this field it will provides a list of all possible entries. When the entry you want is highlighted, press "." to select a sub-field or "Enter" to finish.

Expression properties that represent **Time Values** now show an additional list of time units to choose from. For these properties, the default time unit is hours (Simio's "native" time unit), but you can select days, minutes, or seconds, and Simio will convert the value internally. This currently applies to time properties on the standard library objects, as well as the DelayStep, ActivityElement, and TimerElement.

Enhanced **hierarchical element referencing** in expressions (for example Server1.InputBuffer.MemberQueue.NumberInQueue) and foreign expressions (searching across objects).

Library Objects and Steps

Renamed the Item library object to **BasicEntity** to better reflect its lineage.

Added a ***TravelerCapacity*** property on links. This makes it very easy to model a "spur" as a bi-directional link with *TravelerCapacity* set to "1"

Enhanced **DesiredDirection** property to permit better flow control logic. Setting [LinkName].DesiredDirection = Enum.TrafficDirection.None will close off new entry onto the link. This works for any link (uni-directional or bi-directional).

Added a new **AccumulationLength** state variable on links. Combining this with the Monitor element below, allows the creation of sensors for conveyors.

Added a new **Monitor** element for monitoring a discrete or crossing change of a state variable. This can be used for discrete and continuous modeling. With Dennis' new AccumulationLength state variable on a link, you can now build 'Conveyor Sensor' examples.

Improved behavior of **Connector** object (and zero-travel time **TimePath**) so these objects should work better with a wide variety of situations.

Added **more add-on process** triggers to library objects to provide additional opportunities to supplement standard logic with user-written logic.

Revised **automatic statistic collection**. Now all queues that have statistics enabled are automatically reported. However queues with no activity are suppressed to keep reports shorter.

Simio Sprint 14 –December 26, 2008

This was a somewhat small release due to other staff commitments. Our primary focus for this release was reports, transporters, and miscellaneous small improvements and cleanup.

Reports

While the pivot table provides some easy to use and interesting output data mining capabilities, we plan to also include more **traditional reporting capability** as well as the ability to create your own reports. As a first step, we have changed the Results view to include tabs for Pivot Table and **Reports**. The Reports tab contains an example report. In future releases, the default report(s) will be enhanced as well as the report design feature will be added.

Added **Print Preview** ribbon to support **printing and exporting** of both styles of output results.

Transporters and Entity Movement

Implemented design changes to make modeling **vehicle turn-arounds** and spurs easier. A turn is handled as a 180-degree "pivot" of the entity's leading edge at the node.

Added **Traffic Direction Rule** to Links. This rule can take on one of three values:

- First In Entry Queue – The order that entities are allowed entry onto the link is dictated by the link's Entry Ranking Rule (FIFO,LIFO,SVF,LVF).
- Prefer Desired Direction – Entities that are not traveling on the desired direction are only allowed entry on the link if no entities are waiting to go the desired direction. If there are multiple candidates that can enter, then order is dictated by the Entry Ranking Rule.
- Match Desired Direction – Entities are only allowed entry on the link if their desired direction matches the link's DesiredDirection state value. If there are multiple candidates that can enter, then order is dictated by the Entry Ranking Rule.

Added **DesiredDirection State** to Links. This state variable may be assigned the values None=0, Forward=1, Reverse=2, Either=3. If this state variable is set to None, then no entry is allowed onto the link (applies for any of the Traffic Direction Rules).

Miscellaneous

Improved the GUI in the Experiment view.

Associated the Simio Project Files (.SPF) with Simio, so now they show a Simio specific icon in windows explorer, and you can double click to open them.

Updated the way we do selection in the facility view.

Added **support for lower-end 3D hardware** so 3D features will perform correctly on more (and older) systems.

Tweaked the Process/Elements views so that the ribbon reflects the view you are currently in.

Enhanced the behavior of the auto-complete feature (similar to Intellisense). Unfortunately, we uncovered some technical problems which will seem to prevent this approach, so we will probably implement a new approach in the next sprint.

Added the following math functions: Sin, Cos, Tan, ASin, ACos, ATan, Sqrt, Log, Log10, Round, Floor, Ceiling. We already had Abs, Min, Max, Pi, and E. These functions can be accessed in an expression field by using the syntax **Math.*FunctionName*(*Arguments*)**

Simio Sprint 13 –December 5, 2008

Big News 1

We are including Orthographic and Perspective 3D animation in Simio Design Edition at no extra charge. We feel that it is so tightly integrated that it provides unprecedented ease of use suitable for everyone. And you can easily switch between the three views as needed so you can watch or work in the best view for what you are doing.

Big News 2

One of the biggest problems of using 3D animation on a routine basis is that it is hard to find or draw the symbols you need. No more! Setting a trend that is sure to be copied, **we have tightly integrated a link to Google 3D Warehouse**. With a few simple clicks you can quickly search hundreds of thousands of 3D symbols, find just the right one, and then embed it in your model or your own library. Simply look for the **Get 3D Symbol** phrase when you don't find what you like in the included 3D symbol library.

Our primary focus for this release was to clean up the UI and address dozens of nagging problems to make modeling cleaner and more robust and make the product easier to use and demonstrate.

Vehicle object behavior and link movement

A thorough design review & **refinement of the Vehicle object design** was completed and implemented.

This involved a review of several steps critical to the Vehicle's process logic, including the Pickup, Dropoff, PlanPickup, SelectVisit, Ride, and SelectDropoff steps. These steps are used by the Vehicle object and the TransferNode object to implement behavior such as selecting a vehicle for an entity to ride on, to pickup and dropoff entities, and for the Vehicle object to plan and select its next pickup or dropoff destination.

The design effort of the Vehicle object involved also adding two new property-related to enhance the design. First, a new **TaskSelectionStrategy** property now allows an end-user to specify how a Vehicle chooses its next pickup or dropoff task if multiple options are available. The available rules are Smallest Distance, Largest Distance, Smallest Priority, or Largest Priority. Second, the **IdleAction** property for an 'On Demand' vehicle type now has the option of 'Roam' in addition to 'Park At Node' or 'Park At Home'. The **IdleAction** option specifies how a vehicle behaves if it is currently idle, and if the option is 'Roam' then the vehicle will roam around in the system until it either comes upon a valid entity waiting for pickup or a pickup reservation request is made.

Several bugs related to link-based movement by entities/transporters were also fixed during this sprint.

New Demonstration/Example Models

PackagingSystem.spf –This example models a conveyor system, and is a good example of traveling along conveyors that includes the merging of conveyor flows.
PharmacyExample.spf-This example models a small pharmaceutical supply chain, with a single pharmacy demand point that is replenished by deliveries from a pharmaceutical packing plant.
TrafficIntersection.spf-This example models a small traffic intersection, and is a good example of using add-on processes in a model that involve a possible time delay and can thus hold up the entity's progress until the process execution is completed.

GanttExample.spf-This example models a simple manufacturing line using the Workstation object and illustrates the generation & gantt presentation of a simulated operation sequence.
TransportationSystem.spf-This example models the movement of 'People' entities both walking and riding on 'Cart' Vehicle objects. This is a very simple example of entity/vehicle movement along links, and basic ride/pickup on vehicle objects.

Drawing/Symbols/Decorators

There is now a new **import dialog** available when importing external symbols. It allows you to specify size, and orient things correctly before you fully import them as symbols.
A simple **label object** can now be used to label parts of a model.
A **Conveyor path decorator** was added. You can also now clear a path decorator from a line or link by clicking the "Clear Decorator" button on the "Path Decorators" contextual tab.
More objects can now have **skins** applied, and look correct when doing so. Lines and lines or links with Path Decorators can now show a skin applied to them, and show correctly.
A user-definable "Height" for polygon, closed curve, rectangle, ellipse, polyline, and curve was added. Using this functionality will make it easier to draw things like **walls** (a polyline 0.2 m wide by 2 m tall).
The ability to draw **"Attached" queue animation objects** was added. When a simulation object is selected, you can click on the "Animation" contextual tab, and click on the "Attached Queue" button to draw the queue. This queue will now be associated with the object, and in cases where the object is an agent, entity, or transporter, the queue will animate with each realization animating through the system. This can be used to animate things like the "BatchQueue" to visualize what entities are group members of an entity moving through the system.
Properties Window
A **"[Create New]"** entry for element or state reference properties was added. Clicking "[Create New]" will automatically create a new object of the appropriate type, and set the property value to it. Double clicking in the property name column of the property grid for any element or state reference property will navigate you to the view that state or property is in. Double clicking when a property has no value, will create a new object, and navigate you to it in one action. This is useful for quickly defining add-on processes.
Auto-complete changes:

- Now applies only to expression properties.
- Improved content filtering.
- Added icons.
- Added Random, Math, and Run pseudo-objects.
- Made it a little smarter about when to pop up and when not to.

Process View
Several enhancements were made to the **process editor**. Click and drag in empty space in a process to move it back and forth, clickable arrows also appear to the left and right to indicate a process continues in that direction. The steps were moved out of the ribbon and into a left hand panel. You can either click on a step in the panel then click to place it, or drag a step out of the panel and drop it to place it. After you have selected a step to place, we draw it under the mouse cursor until it "snaps" into place.
The elements list was moved into a tab in the process view. The elements were also given their own icons in their list view. The list also indicates which elements are inherited using a green arrow state icon.
The Process view now shows element icons instead of big blue circles.

Experiments

The following changes related to **experiments** were made:

- Each experiment now has an initial scenario (row).
- You can now select a response column to set its name and expression in the Properties window.
- Response columns can now be removed.
- Experiment runs now detect runtime errors and set status accordingly.
- Responses now show average values across replications.

Miscellaneous

Functionality was enabled related to **object sub-classing**. When editing a subclass you can choose to override (replace) or restore a process from a base object. Any processes added, changed, or removed from a base object properly propagate to all of its subclasses as well.

A very simple **auto-rotate** function was added. If you are in perspective view, and hit the "E" key, the model will slowly rotate. Any other action will stop rotation

A "Quick-Add" reference property now copies the current value and default value to new property.

A **Results View** was added so you can view statistics during an interactive run. If the model is running, the statistics in the Results View are automatically updated each time the run is paused or stopped.

An interactive run now uses current values of model's properties. Also, runtime errors now leave you in run mode but paused, so you can look at the trace output.

We made the following changes to list-oriented views (e.g., Interface View, Element tab of Process View):

- Added Details view support (switch between Details and Tile views by double-clicking the icons)
- Made Details the default for some views
- Changed Tile View to use the smaller icons

Simio Sprint 12 – November 14, 2008

This is our most significant release to date. We have made many changes to improve usability and stability.

Objects and Drawing

We improved the concept of relating **physical size** in the simulation with that in the animation. We set the standard length unit of to be one meter. Anywhere you see a unit-less number in the context of length, you can assume it is in meters. This scale is reflected in all symbols in our library as well as all default objects and paths. (Length conversions will be provided in a future release.)

Simulation objects now have resize handles to **graphically resize** them. We also added the ***Size* property** in the property grid to allow you to specify a specific size (length, width, and height). When animating, objects now pick up their size from their Size state variable. So, if you assign a new value to the Size.Length of an entity, you would see it change in the animation.

You can also **rotate** simulation objects by holding down the Ctrl key while dragging a resize handle. And in 3D view you can **change the Z location of an object** by holding the Shift key (to lock X and Y) while moving the object.

We added a new **drawing grid**, which in top down view shows measurement units on the top and right side. This again reflects the meters scale, but you will see the scale dynamically change as you zoom in and out. Using the grid can be very helpful in drawing objects to scale.

We have added **Use Skin** and **Color** buttons to the Draw ribbon to respectively apply patterns and colors to object surfaces. If you put jpg, png, or bmp images inside the Simio\Skins subdirectory you will be able to apply those images to objects in the drawing space. Users can also apply Skins to Path Decorators on lines and links.

Users can now **associate a set of symbols to a single simulation object** and the animation can switch between them at runtime. To do this simply select a simulation object, and click the "Add Additional Symbol" button. This will add another symbol just like the one that is currently shown. You can then apply a new symbol, or recolor the existing one. To switch between symbols at design time the "Select Current Symbol" drop down can be used.
If you wish to assign a **random symbol** from the list of symbols to an object when it is created, just turn on the "Random Symbol" property. To apply a **user selected symbol**, you can provide an expression for the "Current Symbol Index" property. This expression will be evaluated, and the result used to index into the associated symbol list (starting at 0).

User Interface

We updated the overall UI of the product by improving the design of the ribbon menus.
The **View button** is now much easier to use. We collapsed several of our views into a single Data view. The Schedule, List, Table, Material Arrival, Function Table, and Changeover Matrix views are now all contained in that single view, to aid in navigation and use of the product.
We made more **contextual operations**, that is they don't show up until you select something where they would be appropriate. One example is selecting a simulation object will bring up a tab showing the symbols you can apply to it. Another is selecting a line or link brings up a tab showing what Path Decorators you can apply to them.
We also added a **View tab** for the Facility view, to allow you to more easily switch between view types (perspective, orthographic, and top down), and view options (axis, grid, labels).

We added ribbon buttons for toggling the display of the application-level windows (Properties, Errors, Trace, etc.)

Library Objects

We have added **basic statistics** on Entities, Transporters, Transporter Units, Paths, TimePaths, and Conveyors. We also added **Failure Statistics** on all objects that have failures. The Pivot Table has been refined for easier use.

Links are much more stable now. Issues have been fixed with bi-directional links, variable length objects that span multiple links, links that are drawn to scale, and dynamically resizing links during a run.

BasicNode and **TransferNode** objects have many improvements. They now have '**Routing Out Logic**' categories in which you can set an entity's next destination to either 'Continue'. 'Specific', 'By Sequence', or 'Select From List'. Node parking queues were changed to stations to better support capacity and transfer delays into parking. Added 'OnEntered' and 'OnDeparted' standard processes and 'Entered' and 'Departed' add-on process triggers. Simplified names of nodes associated with another object (like those in Server).

We added basic functionality for **Workstation**. WorkStation is the big brother of Server encompassing **multiple Resources**, **multiple Activities**, **Material consumption and production** (including optional use of **Bill of Materials**), **Sequence-Dependent Setups**, **Gantt Chart support**, and more.

Miscellaneous

Experiment window has been enhanced by eliminating properties that are unlikely to be used and adding a progress bar to indicate scenarios being run.

Object building has been improved. The repeating property editor can now use right-click for property referencing. We added repeat group to repeat group mapping and fixed parsing to properly handle function references on foreign objects (e.g. functions on nodes used for dynamic routing).

We have improved Error Handling so problems are handled more appropriately. Unexpected errors will result in a file that can be returned to Simio to aid us in diagnosis.

And finally, implementation has begun for **auto-complete functionality** (similar to Intellisense). This feature will help you select variables and write expressions by providing a list of available items. The feature is still incomplete and has no mouse support, but you may find it useful in some cases. We also started implementation of **Gantt Charts**. You will see the user interface, but not much more yet.

Simio Sprint 11 – October 25, 2008

We now have **Bi-directional Paths** and **TimePaths**. This means that vehicles can travel in either forward or reverse direction on the same path. And we have included some very flexible options to help control vehicle movement. Also, our links now display arrows which indicate their direction.

Vehicles have also undergone some improvements in function and ease of use. You no longer have to place a vehicle on a Network before the model run. And it is no longer necessary to specify an initial node for your vehicle.

Several of the objects in the Standard Library now have a new "**Add-On Process Triggers**" category. Within this category, triggers are executed by objects when certain events occur. For each trigger, you can optionally specify the name of a process defined in the parent model (i.e., an "add-on process") that is executed when the trigger occurs.

We have enhanced the **Trace window**. It is now viewed at the bottom of the application window when the Trace button is selected. Also the Trace feature supports multiple models.

We now have initial support for the **Gantt view**. This does not yet have much functionality, but illustrates a capability that will be available to support scheduling applications.

The **Views** button has been rearranged in order to make it a little easier to use. (Additional improvements are still in the works.)

We have enhanced and updated our **graphics handling** so that we now work as intended on many more systems. If you have previously had any problems with graphical display and those problems still exist, please send us a screen capture along with a description of your OS and graphics card.

We have fixed quite a few nagging bugs that crept into earlier releases.

The **Issues submission form** is updated to make it easier to use and we added a **Suggestion submission form**. Both can be found on the start page of Simio.

There is now a bare-bones help file which we will be filling in future sprints. You can access this file by clicking on the blue and white question mark at the top right-hand side of the application.

We've simplified referencing properties via right-clicking in the Properties grid in the Facility View. This is one step in making object building and experimentation easier.

Simio Sprint 10 – October 3, 2008

We have enhanced and updated our **install** to include the 3.5 version of Microsoft's **.NET Framework**.

We have also added a way to create and use **Path Decorators** which can be applied to a link, polyline, or curve. Also, it is now possible to define a set of cross sections to extrude along the path, which also applies repeated sets of geometric shapes along a selected path. "Track" and "Road" have been conveniently provided as **examples**.

We have added the important ability to generate **Network** elements via the Process View. We have enhanced this feature with the added benefit of **adding links** to and **removing links** from networks via the Facility View.

We have made considerable improvements to how **Time-Based Failures** are currently executed as well as adding them to Vehicles and Conveyors. And we will continue to improve upon Failures for the next sprint which will include Count-Based Failures.

Initial support has been implemented to support **repeating properties** in the **Interface View**. In the **Properties** grid, we have added the functionality of right-clicking on a property to map it to a property of a containing object. In other words, you can associate additional properties to upper level objects in your model. In the **Table View**, we have added the ability to add to and remove from the Job Table a single embedded sequence table.

We have updated the design and properties of the **WorkStation** object (renamed from Workcenter). We do not yet have much execution behind it (hence the name starting with '_') but you are invited to review the properties and descriptions. WorkStation will be the big brother of Server and will encompass **multiple Resources**, **Operations** consisting of **multiple Activities**, **Material consumption and production** (including optional use of **Bill of Materials**), **Sequence-Dependent Setups**, and much more.
It is important to note that we have included two documents of interest. One of these documents is the ***Known Issues.pdf*** which lists ways to work around some road bumps you may encounter. The other is the ***Simio Feedback Form.xls***. By submitting this document to support@simio.biz, you can provide us with valued information about your experience with Simio. Both of these documents can be accessed from the Start Page when you launch Simio. They can also be found in the zipped folder that contains the Simio installation files.
We have also made significant enhancements to ***Getting Started with Simio.pdf***. It now includes instructions for using Nodes, Links, Networks, and Vehicles, and also includes an appendix describing all of the standard library objects and their properties.

Simio Sprint 9 – September 12, 2008

We have done a great deal of internal work regarding **how resources are defined, used, and reserved**. We have simplified the concept of resources to eliminate the resource types. And while any object can still be a resource, we changed the internal representation so that objects do not carry the overhead of being a resource unless the object-builder specifically enables that functionality.
We have also added a significant new capability to **reserve resources** for use at a future time. This applies not only to a single resource and a single activity, but also to an entire set of resources required across a series of activities. And the duration of the activities are scaled based on the availability of the resources. This capability will make it dramatically easier to model complex situations involving multiple resources potentially following independent schedules. It also makes it possible to easily handle many "look ahead" situations (for example activities that have make-span constraints) and smarter simulation-based scheduling algorithms. Note that this capability is not yet fully implemented.
We have enhanced how **materials** work. You can specify simple materials or a Bill of Materials (BOM) that describes the sub-components of a material. You can produce and consume integer or fractional quantities of material or of the BOM.
We have enhanced **timers** to allow considerably more power and flexibility. A timer can now fire based on a time interval, an arrival pattern, time in a state, or an event count. Times may also be reset and dynamically disabled/enabled. Among other benefits, the new timer features will allow very efficient and flexible implementation of object failures. Although not yet implemented, objects will be able to have multiple types of failures (count-based, time-based, time in state-based, event-based), multiple failure streams, and the failure streams may be specified internal or external to the object definition.
You will find several new and revised items under **Views**. We moved the **List Data**, **Changeover Matrix**, and **Material Arrival** editors into their own views. A user can create a list of strings or objects. They can reference a list of strings and define a setup matrix of values for changing from one value in the list to another. A user can also define a list of material arrivals, giving the material, the arrival date time, and an expression for the amount to arrive at that time.
We added **Table View** to define custom tables of data. It is preloaded with the model's Job Table, and you can add one or more Sequence Tables. Within either type of table you can add properties to define columns in the table (similar to the Interface View). You can also add "related" Data Tables to any other table.
We modified **Schedule View** so that recurring schedule items now properly display and can be assigned different colors for easy identification.
The movement of entities on **links** was revamped to improve efficiency and support accurate passing of entities of differing sizes.

Simio Sprint 8 – August 24, 2008

We have created a formal **Windows Installer** (msi) to make it easy to install Simio. We are still distributing the files in a single zip file for convenience of download. After downloading and extracting the .zip file to a temporary location on your hard drive, all you are required to do is run the *setup.exe* file – any additional software requirements will be automatically installed.

We have added some important features for **building libraries**. Recall that a model is an object and that a project is a collection of models (or objects). Hang on tight… Since a library is a collection of objects, your project is a library. (Sure, it's OK to re-read that a few times. ☺) As you build models and projects you are actually building objects and libraries which can be reused and shared with others. This makes building libraries dramatically easier.

It is now possible to view a list of your models in what is called the **Project Library**. This list can be viewed on the left hand side of the model window. Click on the Project Library tab under the **Standard Library** list to view your list of models. From this library you are able to drag them into facility views the same way you do any other library. An additional related feature allows you to copy an object from a project opened as a library into your active project by dragging and dropping the object from the Project Library list into the **Project View**. A model can now become a subclass in your active project by right clicking and selecting the **Subclass Model** option. This provides add-on behavior to an existing object while keeping its functionality. And you can drag and drop objects created in your Project into other objects in the same Project.

Be sure to read the *Getting Started with Simio* User's Guide which has been updated to include how to access the Project Library.
You can preview the new **Schedule** view framework that lets you create and define repeating schedules and exception schedules. These can then be combined to form composite schedules. The Schedule view can be accessed by selecting the Schedule View option from the drop down **View** menu. Note that this view is not yet able to interact with models at this time.

An enhancement has been made to the **Process View** so that you are able to view, edit, create, and remove **elements** from your model.

The **Pivot** table has been revised with minor changes to the way it appears and additional **automatic statistics** are now available.

Simio Sprint 7 – August 1, 2008

A revamped **Process View** is available. You can add new processes by clicking "Create Process" on the Process contextual tab. You can add steps by clicking on the step name in the "Steps" contextual tab, then clicking inside the process where you want the step to go. When you are choosing a place for a step, the existing steps inside the process will reconfigure themselves to show what the process would look like if the step was added. You can move a step within a process by clicking on it and dragging it. Finally, you can reroute connections from step to step by first selecting the source step for the connection you wish to re-route, then clicking and dragging the handle at the end of the connection.
The Embedded Library has been updated.

- **Node** and **TransferNode** are complete and available for standalone use. They are also embedded in many existing objects including Source, Server, Combiner, Separator, and Sink. Since a click on a node symbol initiates a link, to access the properties of a node you must control-click on the node symbol.
- **Vehicle** has also been added with significant new functionality. Unfortunately the old Cart and the new Vehicle have a high interaction with the new nodes, and we still have many areas which have problems not yet resolved. These will be addressed in a future sprint.
- **TimePath** is now complete. TimePath is a type of link that specifies a minimum travel time. The ability of entities to pass or block each other is an option.
- **Conveyors** are also complete, providing a single construct with the flexibility to easily model many different conveyor types including both accumulating and non-accumulating conveyors.

We have added **Automatic Statistics** on the base intelligent object (resource-related statistics) and station element (throughput-related statistics) which will show up in many other objects. We have also added a few selected statistics on other objects. The purpose of the statistics included so far is mainly to exercise the experiment and results capabilities (described below) and to help us evaluate the types of statistics that are appropriate. Comprehensive statistics will be provided in a future sprint.
The **Experiment Designer** has been enhanced. After you run an experiment, you can go to the View icon on the ribbon and select results. A new **Experiment Results** window will display the output data in the form of a pivot table. If you have experience with pivot tables (e.g. like what is available in Excel) you will find that they provide a very easy way to manipulate, summarize, and display data and they provide a powerful mechanism for exploring the data (data mining). Highlighting an individual or set of data values will cause them to be **graphically displayed** in the lower part of the Experiment Results window.
We have added an **Interface View**. While this does not yet have much functionality, eventually this will become the place you go to define and customize model (object)-level properties, states, and events.
And finally, you will find new and updated sections in the ***Getting Started with Simio*** **User's Guide** that describes the basics of Process Editing, Dashboard View, Experiment Designer, and Experiment Results.

Simio Sprint 6 – July 11, 2008

A new **Dashboard View** allows you to place several objects which can interact with your running simulation. A Status Label evaluates an expression and displays its current value. A **Status Plot** allows you to plot a single expression value over time, and a Button allows you to **fire** a simulation event arbitrarily in the middle of a run. We have added example dashboards for the Source, Server, and Sink objects. To open them, right click on an object and select "Open Dashboard".

There is now an initial framework for running **experiments**. You can define an experiment on the active model by clicking the new Experiments tab, right-clicking in the Experiments view and selecting New Experiment. In the new **Experiment Designer** window, a column is displayed for every top-level property of your model. To create various scenarios, enter initial values for these properties, and enter the number of replications desired for each scenario. Finally, to run the experiment, select the Experiment ribbon tab and click Run to run the replications in the background while you can continue to work interactively. At the moment the choice of output measures is very limited – this will be enhanced in coming releases.

We have again made minor enhancements to the properties windows.

We have completed functionality in the following library objects: **Source**, **Sink**, **Server**, **Resource**, **Combiner**, **Separator**, **Entity**, **Connector**, and **Path**. You will find lots of new functionality supported here including the full implementation of the new "every object is a resource" paradigm described in the Sprint 5 release notes and support for failures. We are still enhancing Node and TransferNode functionality which will add additional features to some of the above objects. Full Conveyor and Vehicle support will be added soon.

And finally, you will find a new section in the *Getting Started with Simio* User's Guide that describes how the existing Process Editor works. (Note that we expect this functionality to change soon, but you can use this to learn a bit about the capabilities.)

Simio Sprint 5 – June 20, 2008

We have added the ability to place user defined static symbols in your facility view. Clicking on the "**Symbol**" button in the "**Drawing**" group of the "**Drawing**" tab will allow you to place a static symbol placeholder. Once your static symbol placeholder is placed and selected, you can then go to the "Symbols" group of the "Drawing" tab to select a symbol to apply to it. Simply click on a picture in the gallery to apply that symbol.
You can create your own symbols inside your project by clicking on the "Add Symbol" button in the "Symbols" group of the "Drawing" tab. This will open a view to edit your symbol. This new symbol will appear in the left gallery in the "Symbols" group of the "Drawing" tab.
If you want to use an **external 3D model** as a symbol, we have added the ability to **import files**. To create a symbol from a file, simply click "Add Symbol" then, in the symbol editing view, select "Import 3D Model" from the "Drawing" group of the "Drawing" tab. Select the file you wish to import, click "Open", then click in the drawing space to place your model. If you want help to place it exactly in the center, hit the 'Q' key on your keyboard to toggle the center lines for the drawing space.
We continued updates to the **Properties windows**. The property grid now has tool tips and its drop-down lists are now sorted. We added "Reset to Default Value" functionality, via "Reset" menu item or Delete key. We are looking for feedback on the new way properties are nested – we thought we would try something different. We enhanced our support for repeating properties including making those changes undoable and providing better interaction with the error window.
We have made significant progress in designing/improving how objects work behind the scenes and in particular, how they relate to each other.

- **Resources** – Any object can be declared as a Unit, FinitePool, InfinitePool, or ScheduleBased resource, and may be seized and released using Seize and Release steps. This applies not only to obvious object like machines and vehicles, but also to workers and even entities. This provides very powerful flexibility.
- **Object Replication & Dynamic Creation/Destruction**– Finalized design for supporting replication of agent/entity/transporter objects. This allows you to specify an initial count (e.g. give me 30 independent transporters with the same characteristics). We also finalized designs for accessing, selecting, seizing, creating, and destroying those replicated objects via process logic.
- **Transporters**–Refined design of steps/features for handling transporter reservations, pick up, and drop offs.

These design changes are only partially implemented in the objects. Note that most of our library objects (like Server) are still prototype objects, not the final ones. We expect to make major progress on them in Sprint 6.

Simio Sprint 4 – May 30, 2008

We added support for **drawing** simple shapes in the **3D view**. We support **Line**, **Curve**, **Rectangle**, **Ellipse**, **Polygon**, and **Closed Curve**. You can select shape color as well as line style. When in the top down view the shapes appears as they do in the standard 2D view, when changing to Orthographic or Perspective view however the shapes take on a 3D look.
We added support for more simulation animation. You can now place a **queue** in the 3D view and see it animating, which means transporters now animate in 3D. Entities now turn in the direction of the current segment when animating on a link in 3D.

We continued updates to the **Properties** and **Error windows**. The new properties window supports undo and redo, and when undoing or redoing it will try to highlight the affected property. The new error window shows error messages for any properties that are in error. Double-clicking on a row in the error window takes you to the offending property. The new error window automatically appears when there are errors and automatically disappears when there are no longer any errors.
Once again, we have made significant progress in designing/improving how **objects** work behind the scenes and in particular, how they relate to each other. This includes the ability to suspend and resume processes (e.g. to implement flexible failures), support for pattern schedules on Source, creating objects of specified types at specified physical/logical locations, and lots of work on seizing and releasing objects of any type and moving objects between physical/logical locations. Again, these changes are hard to detect and have not yet been fully implemented in all the objects.

Simio Sprint 3 – May 9, 2008

We have added basic **3D animation support**. You can now build and animate your models in either 2D or 3D and can switch between the views at will using the View button on the ribbon. Simple use instructions for currently supported features will appear at the top of the 3D window. If you haven't done so already, please note the updated installation procedure. You may need to install some new software in order to see these 3D capabilities.

Although we have made significant other progress in this sprint, they are harder to see. You may notice changes in the **Properties window**. We have converted that from prototype code to production code and have added some new features. However, most of those features are not yet apparent because the objects have not yet been redesigned to take advantage of it.

We have also made significant progress in designing/improving how objects work behind the scenes and in particular, how they relate to each other. Again, these changes are hard to detect and have not yet been fully implemented in the objects.

Simio Sprint 2 – April 18, 2008

We have enhanced our Microsoft Office 2007-style **ribbon interface** in several ways, including the addition of a View button, some Run Setup information, and an animation speed slider bar. This is still far from complete but provides additional illustration of the possibilities.

You will also now find a **Project View window** for viewing, navigating, renaming, and deleting components of your project. We have added initial undo/redo capability, but it is currently applied to only a few actions.

You will see many new and some apparently duplicated objects in the **standard library**. The original objects (e.g. "Source", "Server", and "Cart") can still be used for building working models, but these are remnants of the initial prototype. The new objects that start with "_" (e.g. "_Source", "_Server", and "_Worker") are a mockup of our final objects. These do not have modeling code behind them, but they can be used to evaluate the interface, the object properties, and the modeling power that will be exposed. We have made significant changes to the Properties Window (by default in the lower right) which you will use for editing object properties. We are especially anxious to hear your feedback on the usability of this interface.

Simio Sprint 1 – March 29, 2008

We have started implementing a better UI. In particular you will see that we have replaced our prior toolbars with the new ribbon technology introduced with Microsoft Office 2007. This is far from complete but provides some illustration of the possibilities. You will see a Home panel with Run Control, and simple Draw capability, a Create split-button that allows creation of new models and objects, the Application button (the green triangle in the upper left), quick button support (to the right of the application button. You will also find run status implemented on the bottom left of the window.

We are continuing work on the design of basic objects, the standard library, the steps and elements, and much more. Preliminary designs will soon be available on the Insider's forum.

Simio Sprint 0 (Early Prototype) – March 10, 2008

This early prototype has been created to aid in discussions by giving form to some ideas and concepts. The basic concepts are implemented, but there are lots of features that are missing or incomplete. In particular, we expect to make significant changes to the user interface (UI).

While we are interested in hearing about anything you particularly like or dislike, this prototype will soon be replaced with early "production" software. Its primary use now should be to aid in understanding the context of any questions posed in the Simio blog.