Module 4

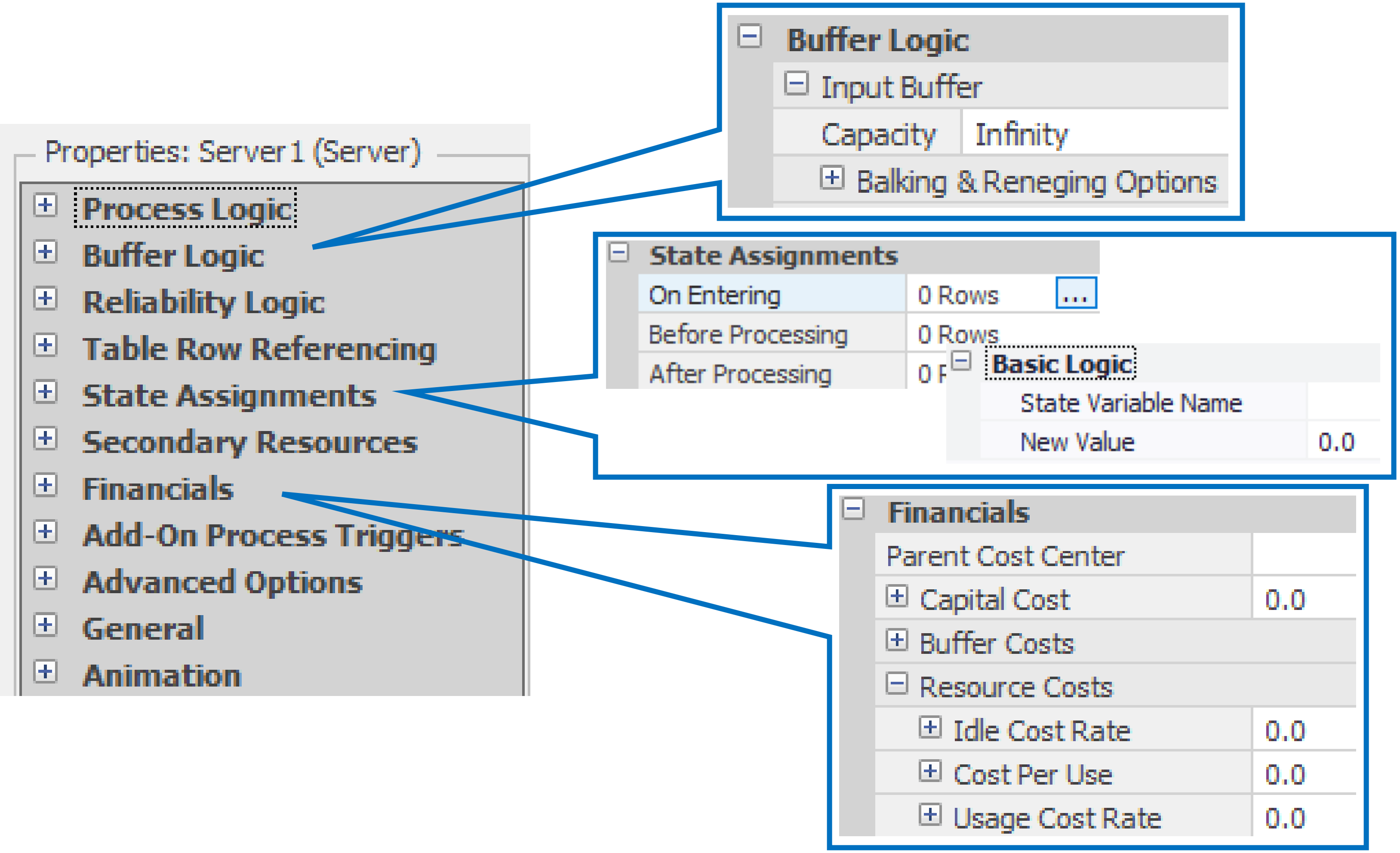
More Simio Library Highlights

Outline – Simio Library

- Overview and the Source Object
- Sink Object and the Source-Sink Model
- Workshop: ConWIP
- Server Object
- Workshop: Agony Airport
- Batching & Separating Entities
- Material Handling with Conveyors
- Vehicles and Workers

Common Object Categories

- Most library objects contain common *categories* of properties - some basic, some more advanced:



Properties: Server1 (Server)

- ⊞ Process Logic
- ⊞ Buffer Logic
- ⊞ Reliability Logic
- ⊞ Table Row Referencing
- ⊞ State Assignments
- ⊞ Secondary Resources
- ⊞ Financials
- ⊞ Add-On Process Triggers
- ⊞ Advanced Options
- ⊞ General
- ⊞ Animation

⊟ Buffer Logic

⊟ Input Buffer	
Capacity	Infinity
⊞ Balking & Reneging Options	

⊟ State Assignments

On Entering	0 Rows ...
Before Processing	0 Rows
After Processing	0 F

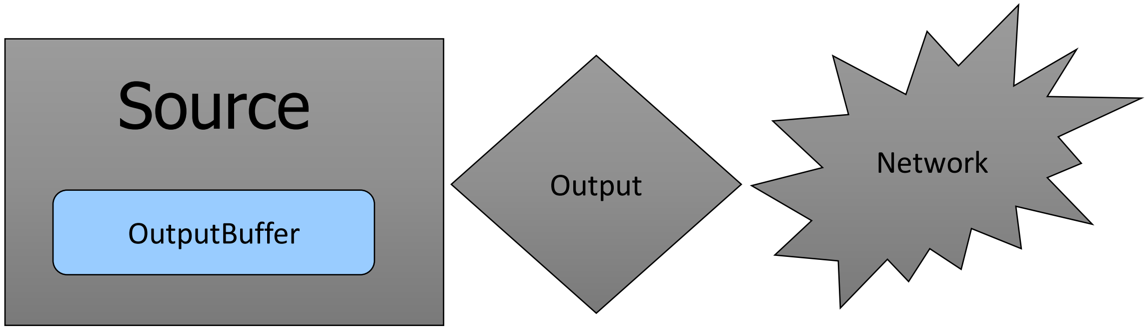
⊟ Basic Logic

State Variable Name	
New Value	0.0

⊟ Financials

Parent Cost Center	
⊞ Capital Cost	0.0
⊞ Buffer Costs	
⊟ Resource Costs	
⊞ Idle Cost Rate	0.0
⊞ Cost Per Use	0.0
⊞ Usage Cost Rate	0.0

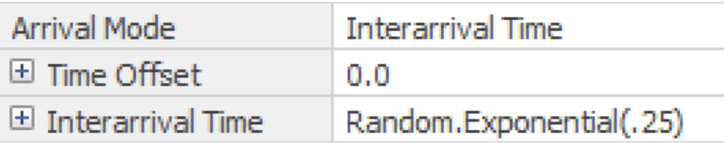
Source



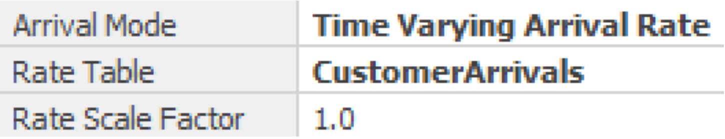
- Dynamic entities enter at the source; they wait in the OutputBuffer station to transfer onto the network or free space at the Output node.
- Major properties include
  - *Entity Type*,
  - *Arrival Mode,*
  - *Entities per Arrival*,
  - *Buffer Logic (balking)*, and
  - *Stopping Conditions.*
- Entity symbols
  - Multiple symbols
  - Can be assigned or randomized
- TIP: Use Source with data table to create multiple entity types

Source – Arrival Mode

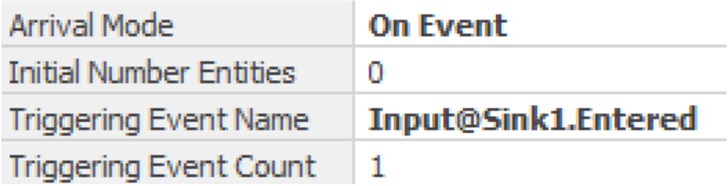
- Interarrival Time
  - Random time between arrivals (typically)
- Time Varying Arrival Rate
  - Uses rate table
- On Event
  - Built-in or user-defined event
- Arrival Table
  - Get arrival times from a table
  - Discrete times
  - Random options



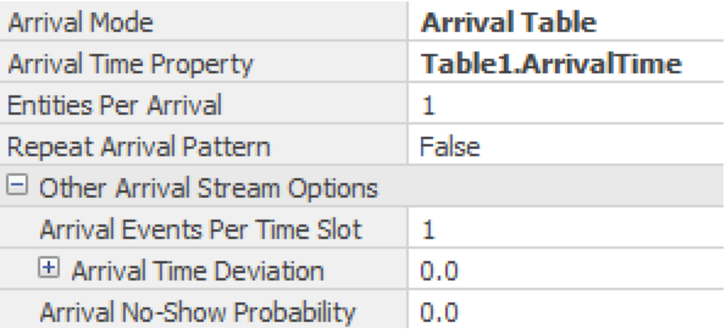
Arrival Mode	Interarrival Time
⊞ Time Offset	0.0
⊞ Interarrival Time	Random.Exponential(.25)



Arrival Mode	**Time Varying Arrival Rate**
Rate Table	**CustomerArrivals**
Rate Scale Factor	1.0



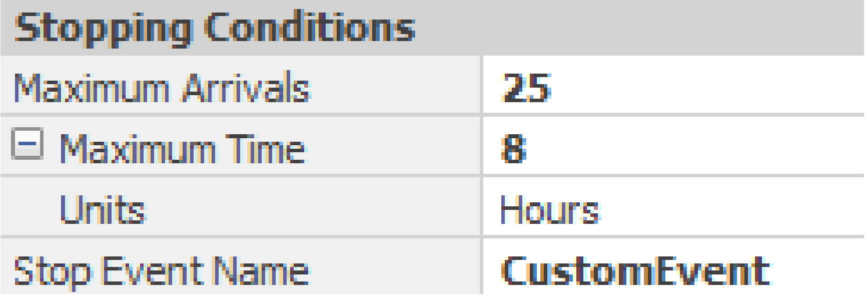
Arrival Mode	**On Event**
Initial Number Entities	0
Triggering Event Name	**Input@Sink1.Entered**
Triggering Event Count	1



Arrival Mode	**Arrival Table**
Arrival Time Property	**Table1.ArrivalTime**
Entities Per Arrival	1
Repeat Arrival Pattern	False
⊟ Other Arrival Stream Options	
Arrival Events Per Time Slot	1
⊞ Arrival Time Deviation	0.0
Arrival No-Show Probability	0.0

Source – Stopping Conditions

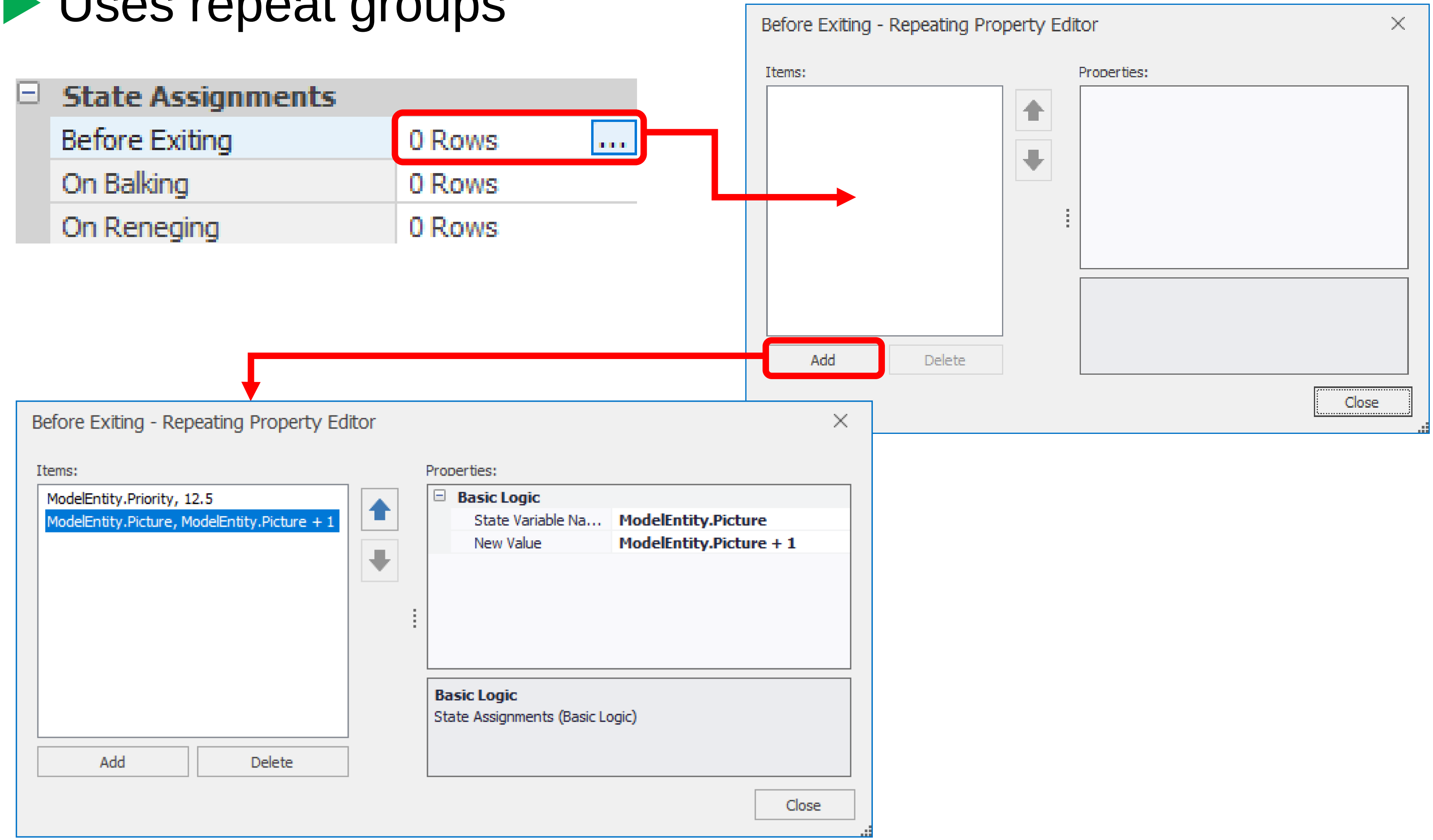
- Determines when you stop creating entities
- Maximum Arrivals
  - Limits number of arrivals
- Maximum Time
  - Stops arriving after specified time
- Stop Event
  - Stops after built-in or user-defined event



Stopping Conditions	
Maximum Arrivals	25
⊟ Maximum Time	8
Units	Hours
Stop Event Name	CustomEvent

Source – Assignments

- Change (assign) the value of a state while model is running.
- Uses repeat groups



State Assignments

Before Exiting	0 Rows
On Balking	0 Rows
On Reneging	0 Rows

Before Exiting - Repeating Property Editor

Items:

Properties:

Before Exiting - Repeating Property Editor

Items:

ModelEntity.Priority, 12.5
ModelEntity.Picture, ModelEntity.Picture + 1

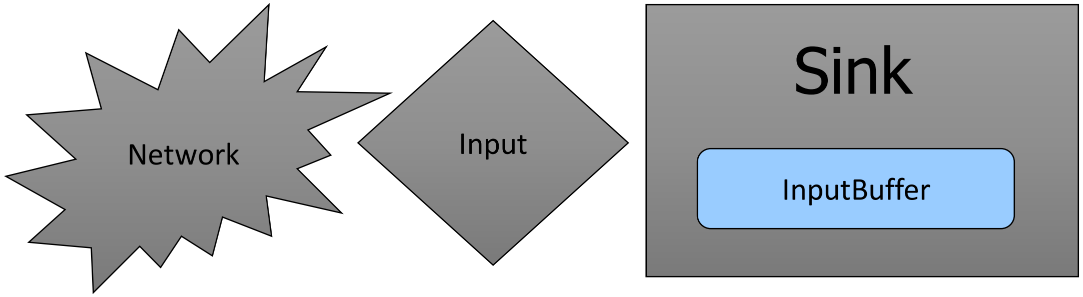
Properties:

Basic Logic

State Variable Na...	**ModelEntity.Picture**
New Value	**ModelEntity.Picture + 1**

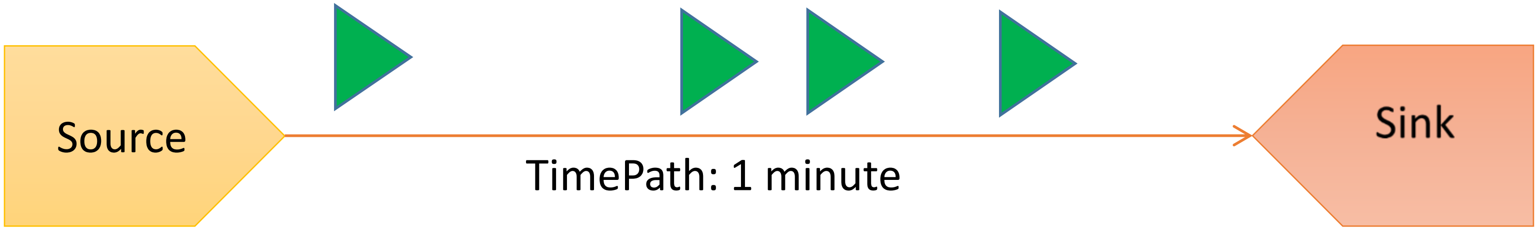
Basic Logic
State Assignments (Basic Logic)

Sink



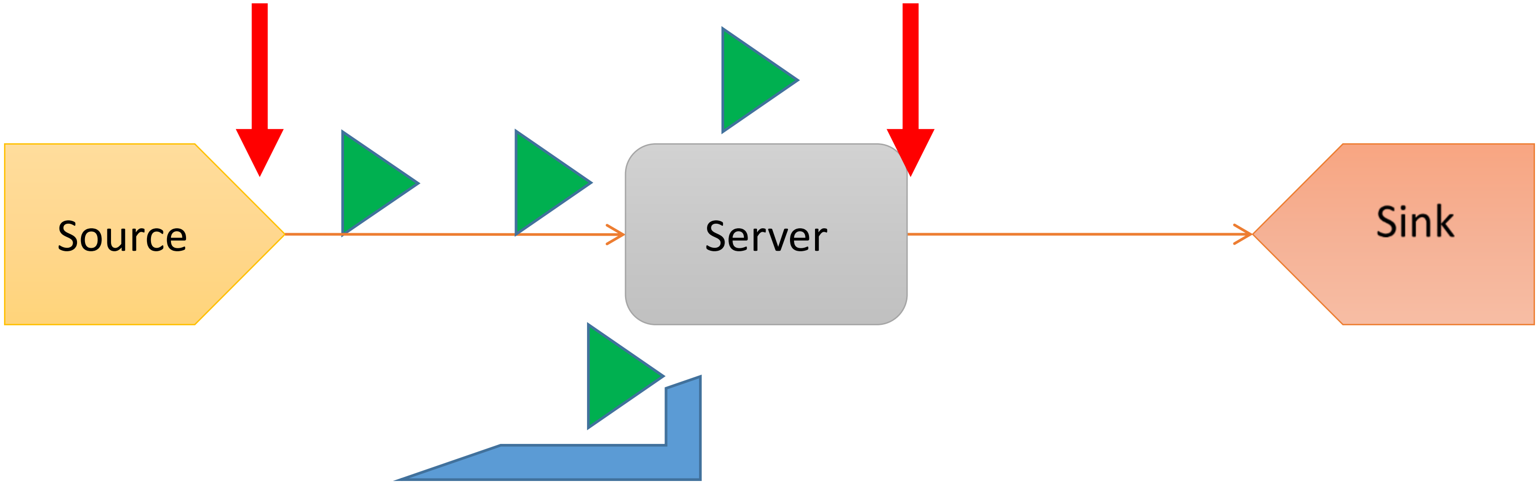
- Dynamic entities depart the system at the InputBuffer of the Sink.
- The Time In System is automatically recorded.
- Optional Assignments

Source and Sink



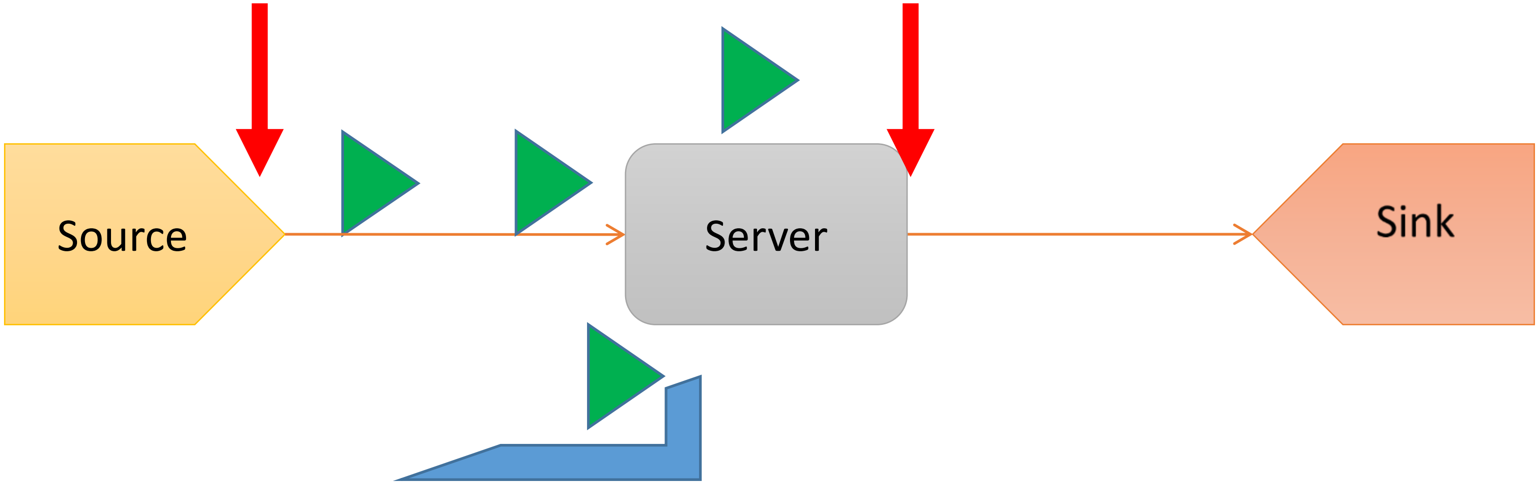
Workshop: ConWIP

- Create a Source, Server, Sink, spaced about 10 meters apart and connected by TimePaths.
- Use all default properties except change Source properties so that a constant WIP of exactly 4 entities will exist between Source and Output node of Server.

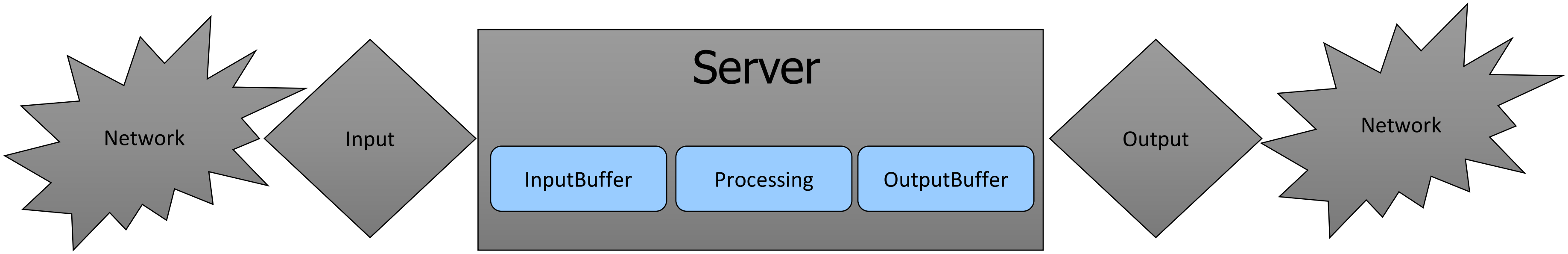


Workshop: ConWIP - Hints

- For ConWIP, need to do the following:
  - Create 4 entities initially
  - Create a new entity when an entity leaves the server
- Source properties
  - Initial Number Entities
  - Arrival Mode - OnEvent

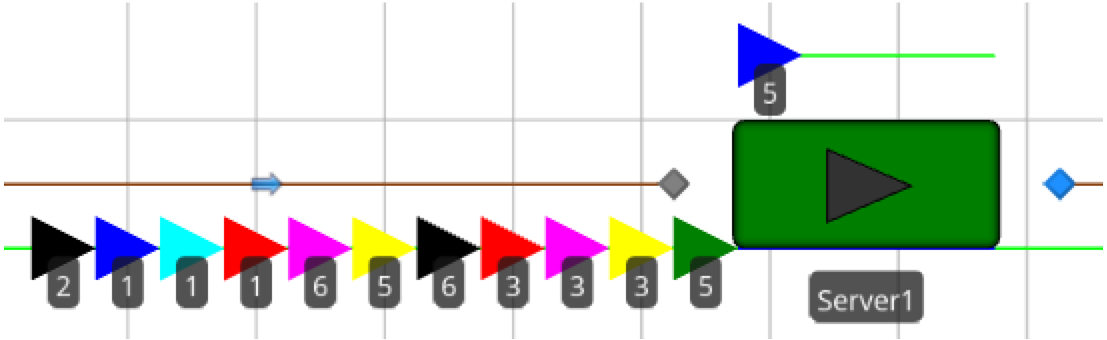


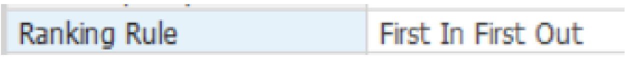
Server



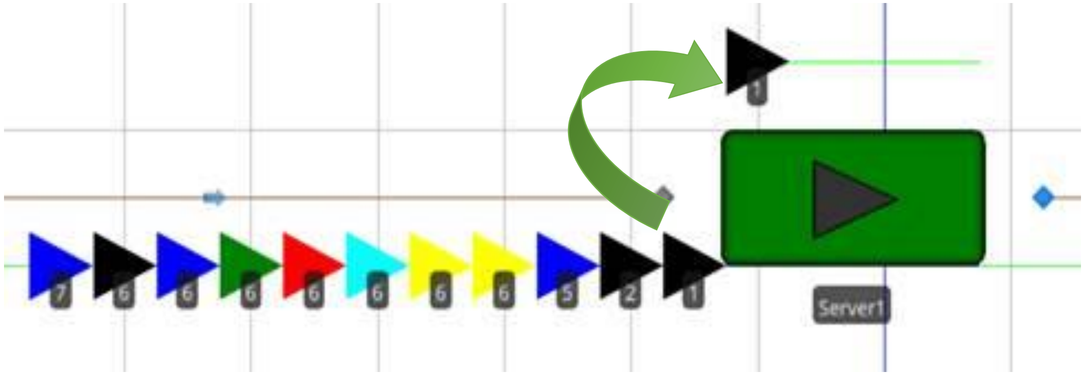
- Arriving entities seize/release the server
- Has a specified fixed capacity or follows a work schedule.
- Ranking of and selection from entities waiting.
- Reliability - failures and repair time.
- Flexible control of secondary resources
- Advanced options to control simultaneous events, reserving, multiple process phases

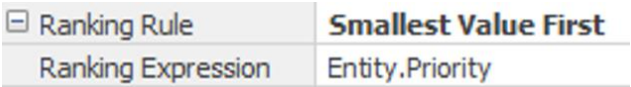
Server – Static Ranking





Ranking Rule	First In First Out

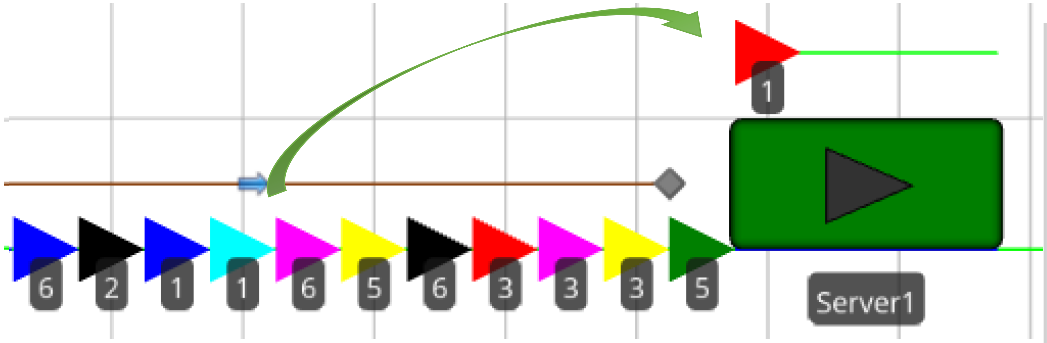


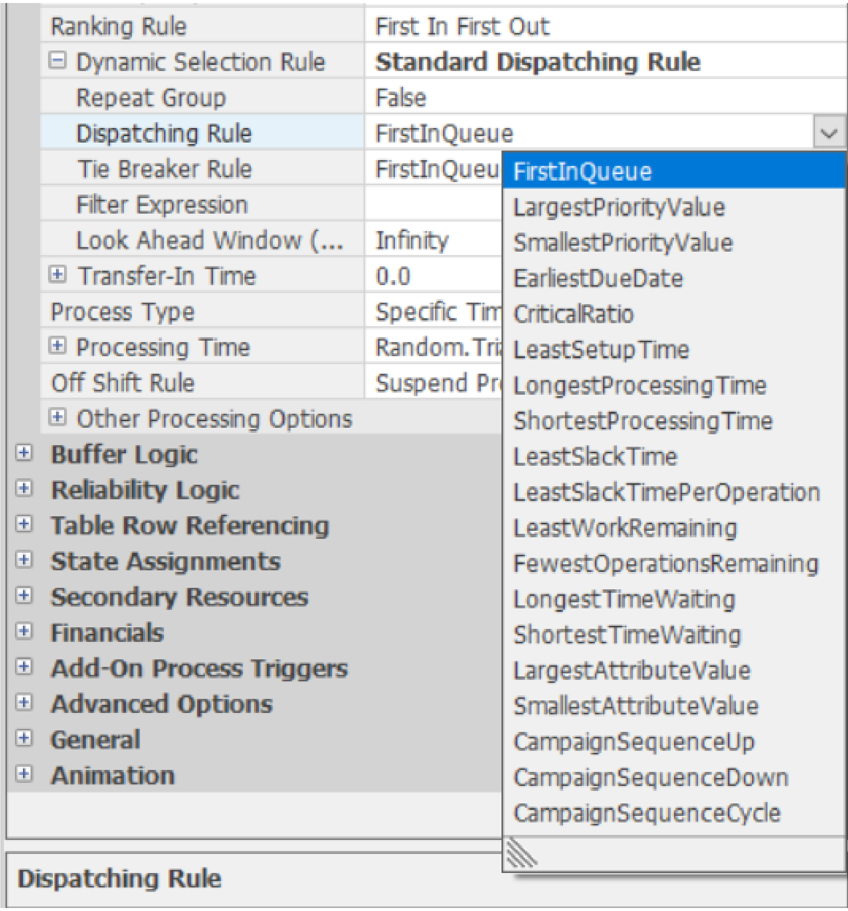


⊟ Ranking Rule	**Smallest Value First**
Ranking Expression	Entity.Priority

- Ranks entities waiting by FIFO, LIFO, or value of any state or expression.
- Unless you choose Dynamic, resource always selects the first member (rank 1) of the queue.

Server – Dynamic Selection





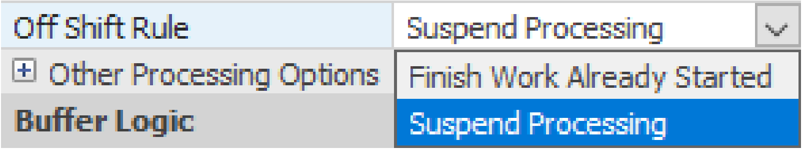
Ranking Rule	First In First Out
⊟ Dynamic Selection Rule	**Standard Dispatching Rule**
Repeat Group	False
Dispatching Rule	FirstInQueue
Tie Breaker Rule	FirstInQueu
Filter Expression	
Look Ahead Window (...	Infinity
⊞ Transfer-In Time	0.0
Process Type	Specific Tim
⊞ Processing Time	Random.Tri
Off Shift Rule	Suspend Pr
⊞ Other Processing Options	
⊞ **Buffer Logic**	
⊞ **Reliability Logic**	
⊞ **Table Row Referencing**	
⊞ **State Assignments**	
⊞ **Secondary Resources**	
⊞ **Financials**	
⊞ **Add-On Process Triggers**	
⊞ **Advanced Options**	
⊞ **General**	
⊞ **Animation**	

Dispatching Rule

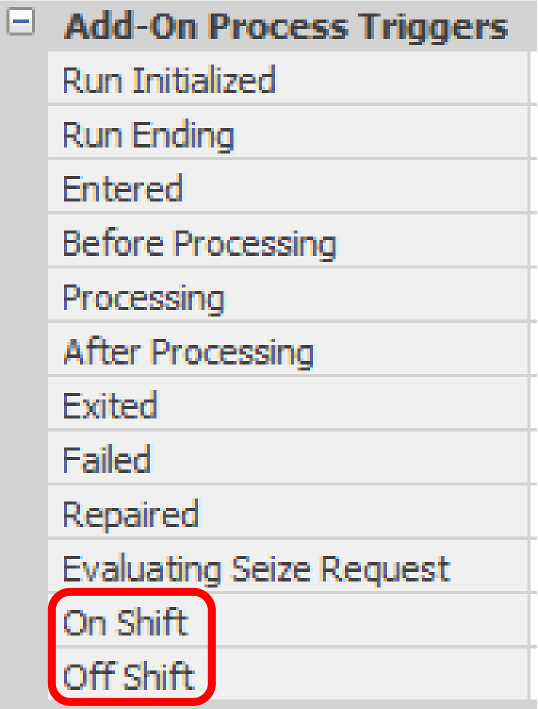
- Dynamic selection looks at all queue members before each allocation to select the "best".
- Rules may optimize locally or globally.
- User-written selection rules.

Server – Off Shift Transition

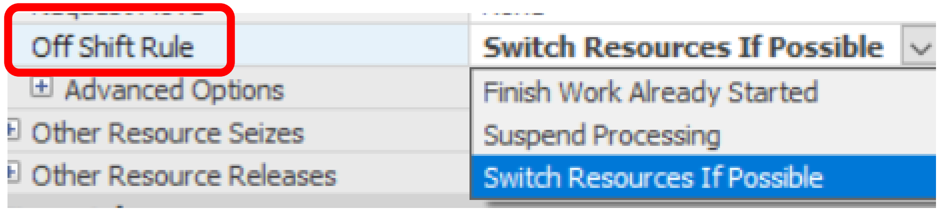
- Off Shift Rule determines what happens to entities in progress when capacity is reduced below the number busy.
- Use Add-On Process Triggers to implement any desired behavior.
- Option to select different secondary resources on capacity (shift) change.



Off Shift Rule	Suspend Processing
⊞ Other Processing Options	Finish Work Already Started
Buffer Logic	Suspend Processing



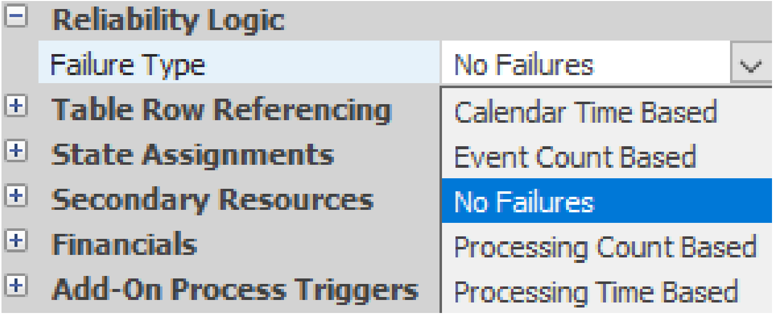
⊟ **Add-On Process Triggers**
- Run Initialized
- Run Ending
- Entered
- Before Processing
- Processing
- After Processing
- Exited
- Failed
- Repaired
- Evaluating Seize Request
- On Shift
- Off Shift



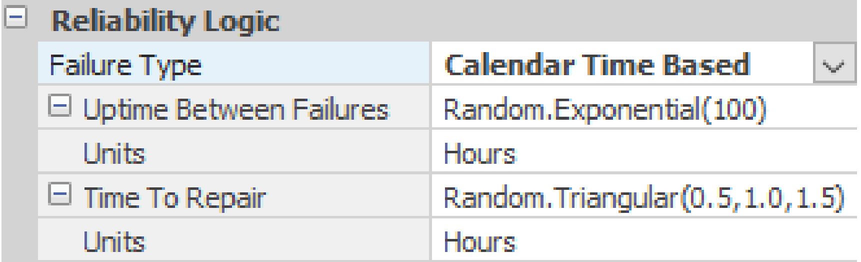
Off Shift Rule	**Switch Resources If Possible**
⊞ Advanced Options	Finish Work Already Started
⊞ Other Resource Seizes	Suspend Processing
⊞ Other Resource Releases	Switch Resources If Possible

Server – Reliability

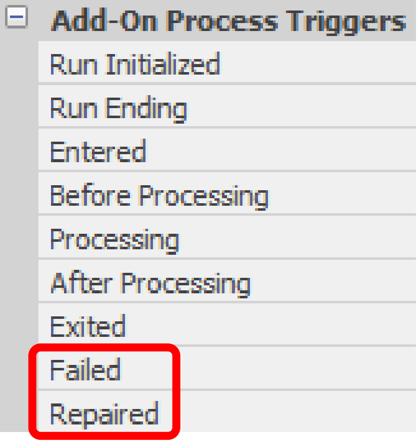
- Used to represent failures
- Four types:
  - Calendar Time
  - Event Count
  - Processing Count
  - Processing Time
- Specify time or count between failures (uptime)
- Specify time to repair (downtime)
- Customize with Processes



Reliability Logic	
Failure Type	No Failures
Table Row Referencing	Calendar Time Based
State Assignments	Event Count Based
Secondary Resources	No Failures
Financials	Processing Count Based
Add-On Process Triggers	Processing Time Based



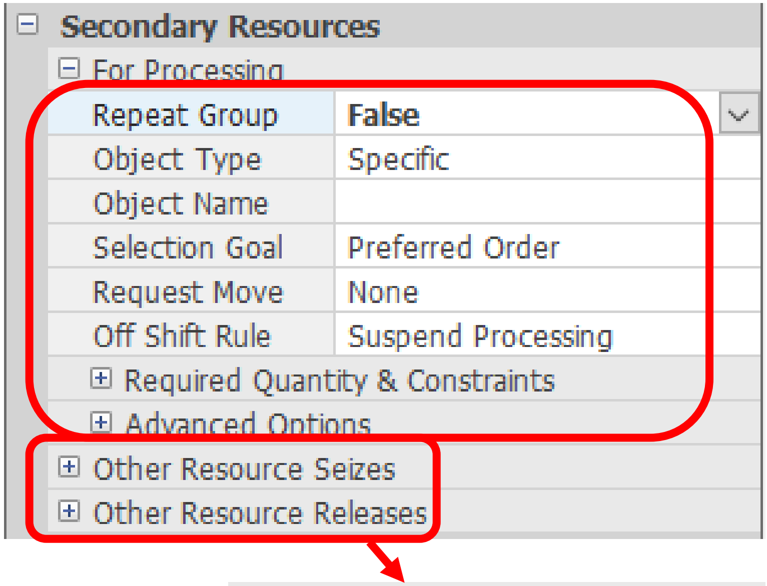
Reliability Logic	
Failure Type	**Calendar Time Based**
Uptime Between Failures	Random.Exponential(100)
Units	Hours
Time To Repair	Random.Triangular(0.5,1.0,1.5)
Units	Hours



Add-On Process Triggers
- Run Initialized
- Run Ending
- Entered
- Before Processing
- Processing
- After Processing
- Exited
- Failed
- Repaired

Server – Secondary Resources

- Resource for Processing
  - Single resource used for entire "Processing Time"
  - Specific or Select from List
  - Optional Repeat Group
  - Optional Move to Location



⊟ Secondary Resources

⊟ For Processing

Repeat Group	False
Object Type	Specific
Object Name	
Selection Goal	Preferred Order
Request Move	None
Off Shift Rule	Suspend Processing

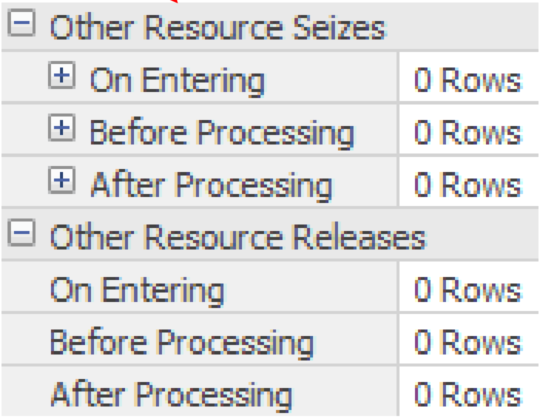
⊞ Required Quantity & Constraints

⊞ Advanced Options

⊞ Other Resource Seizes

⊞ Other Resource Releases

- Other Resource Seizes/Releases
  - Select timing within process
  - Repeat groups for multiple resources
    - (e.g., a worker and a fixture)
  - Seize in one server and release in another provides high flexibility



⊟ Other Resource Seizes

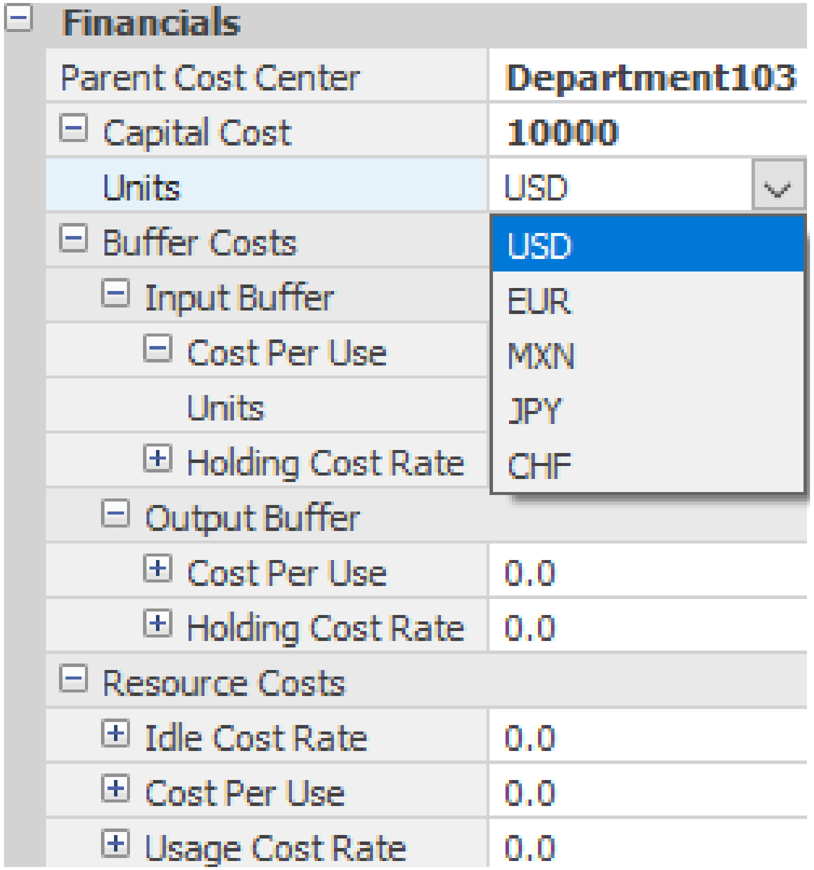
⊞ On Entering	0 Rows
⊞ Before Processing	0 Rows
⊞ After Processing	0 Rows

⊟ Other Resource Releases

On Entering	0 Rows
Before Processing	0 Rows
After Processing	0 Rows

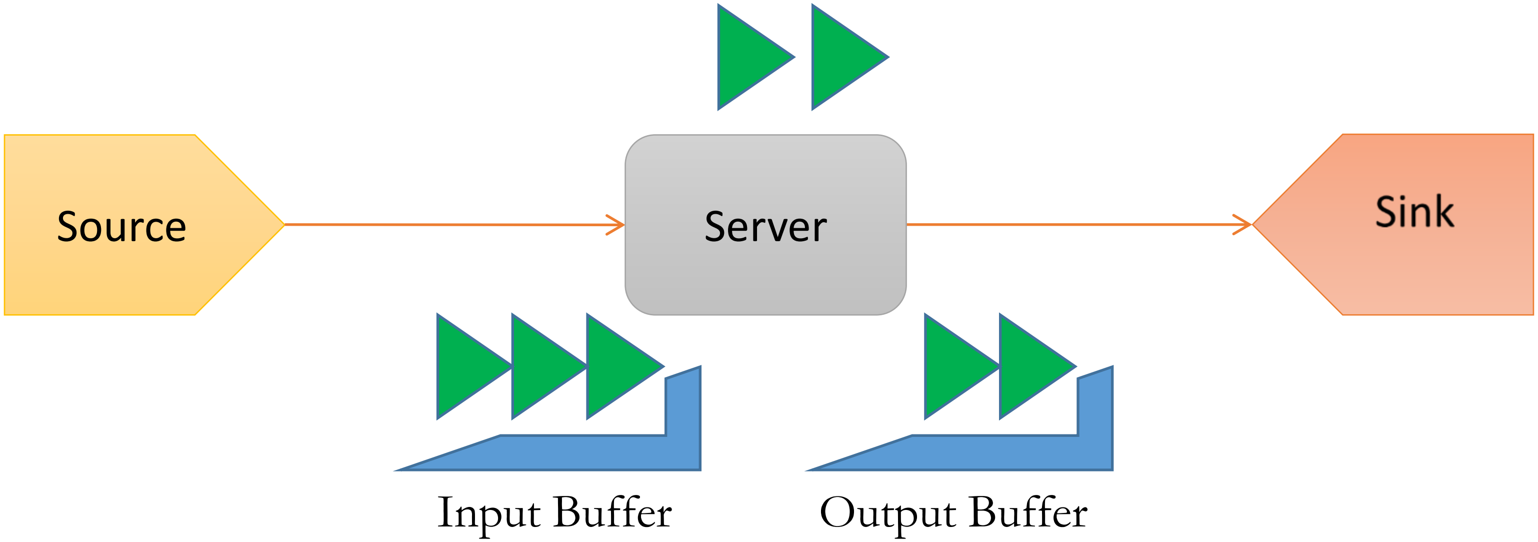
Server – Financials

- Comprehensive Activity-Based Costing
  - Capital Costs
  - Idle Costs
  - Busy Costs
  - Holding/Usage Costs
- Supports all world currencies
  - You supply desired exchange rates
- Unlimited roll-up to higher cost centers



⊟ Financials	
Parent Cost Center	Department103
⊟ Capital Cost	10000
Units	USD
⊟ Buffer Costs	
⊟ Input Buffer	
⊟ Cost Per Use	
Units	
⊞ Holding Cost Rate	
⊟ Output Buffer	
⊞ Cost Per Use	0.0
⊞ Holding Cost Rate	0.0
⊟ Resource Costs	
⊞ Idle Cost Rate	0.0
⊞ Cost Per Use	0.0
⊞ Usage Cost Rate	0.0

Source Server Sink



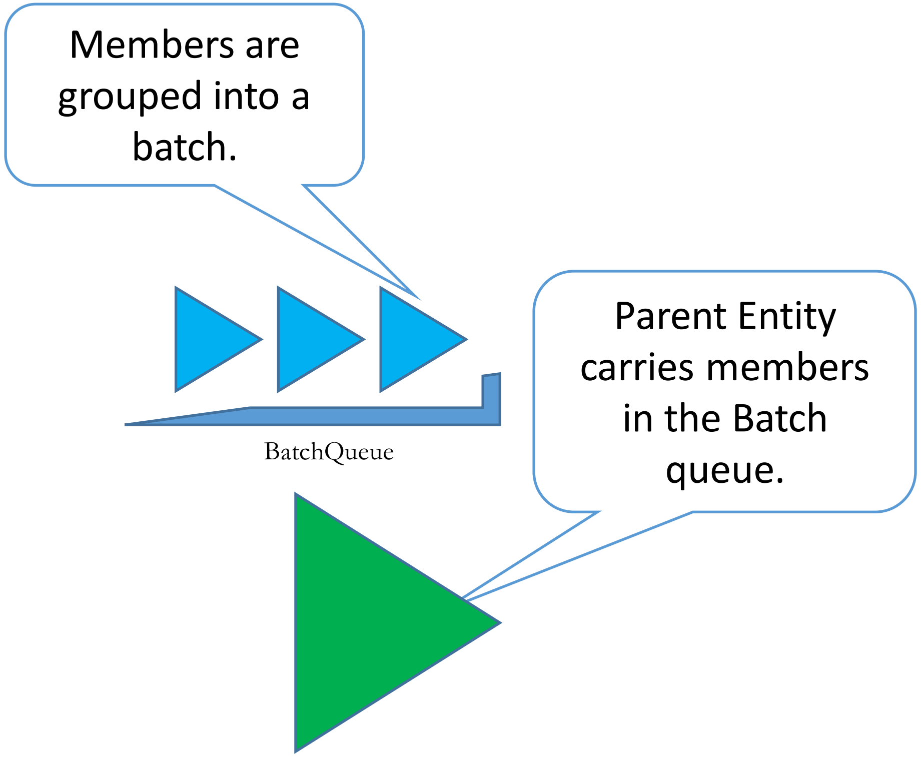
Standard Library
Fixed Objects

Batching

- Parent vs. Members
- Combiner and Separator Objects

Batching

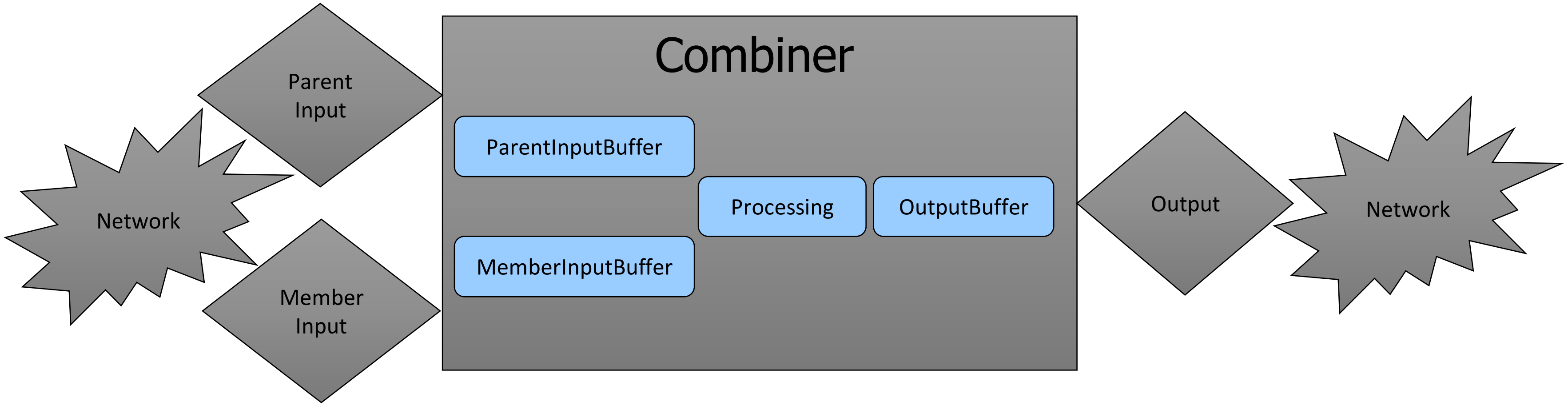
- A group of *member* entities can be batched and carried by a *parent* entity.
- Batched members can be individually split from the batch.
- Parent entities may also be batched.



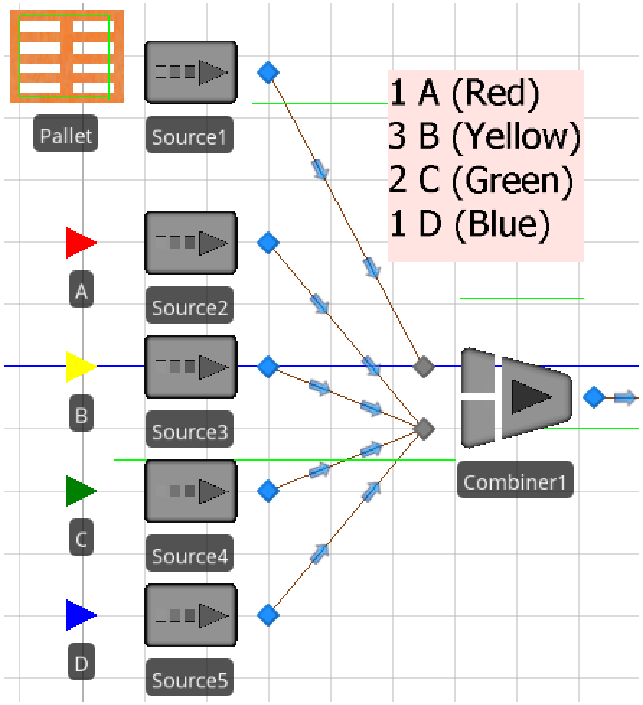
Members are grouped into a batch.

Parent Entity carries members in the Batch queue.

Combiner

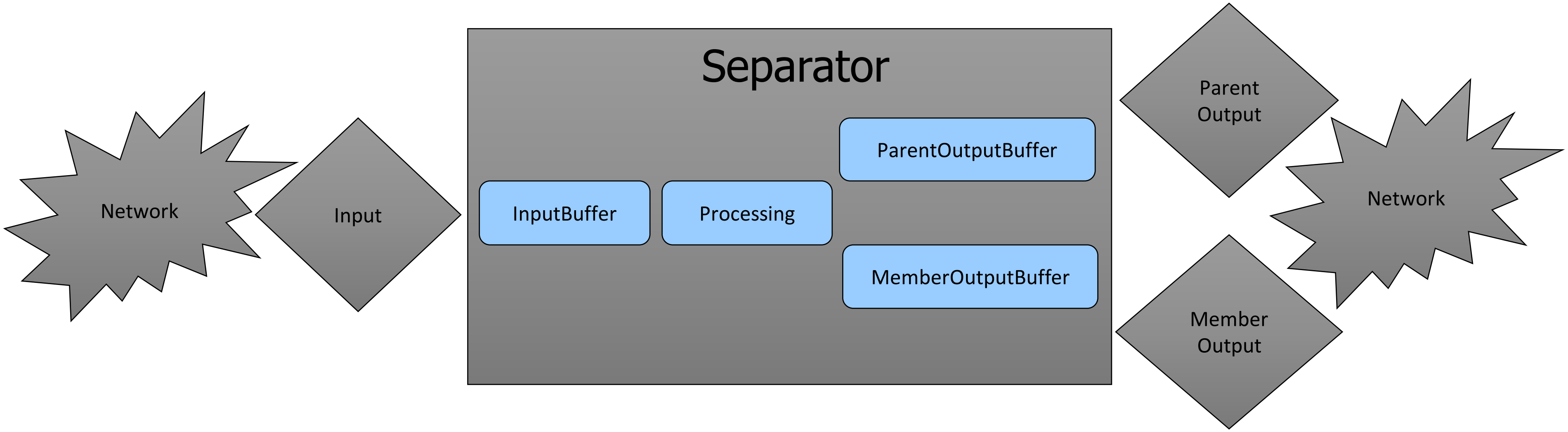


- The Combiner is a Server with batching logic.
- The Combiner has separate input nodes for the parent and member entities.
- SimBits:
  - Regenerating Combiner
  - Combine Multiple Entity Types onto Pallet
  - Combiner Releasing Batch Early
  - Combine Matching Members



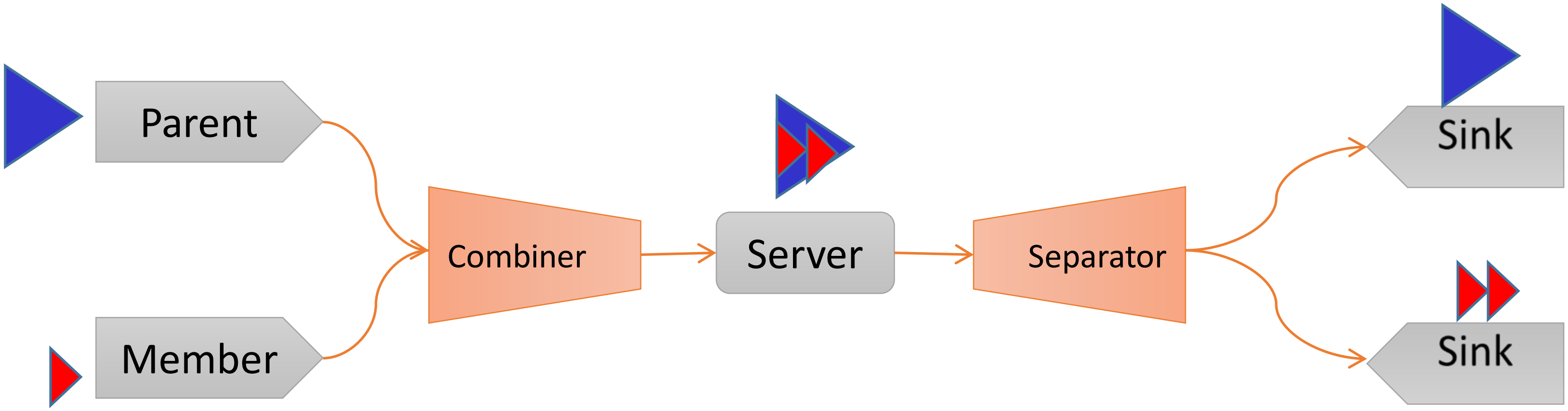
1 A (Red)
3 B (Yellow)
2 C (Green)
1 D (Blue)

Separator



- The Separator has separate output nodes for the departing parent and member entities.
- Member entities can be split from the parent or cloned from the parent or another entity type.
- The Separator is a Server with splitting/cloning logic.

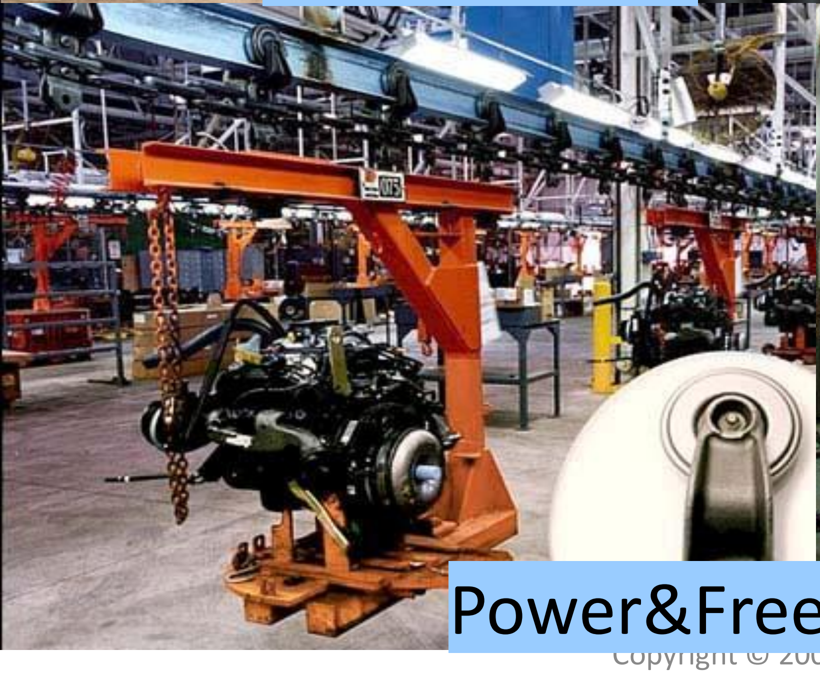
Combine then Separate



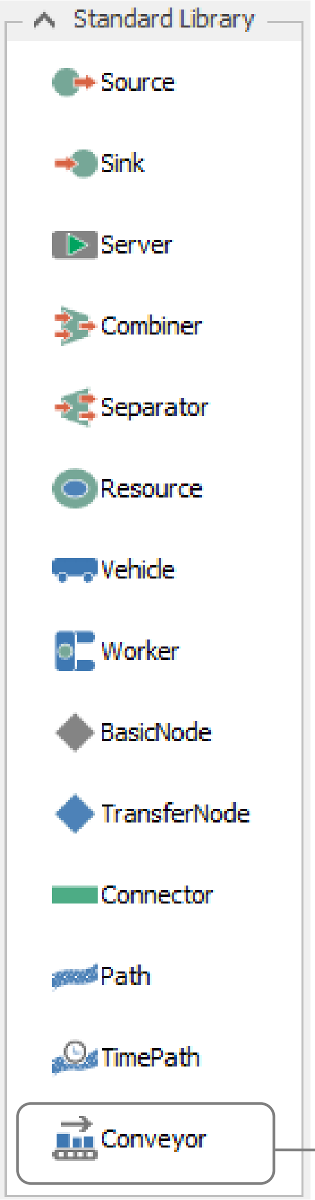
Material Handling

Conveyors

Many Conveyor Types



Material Handling using Conveyors



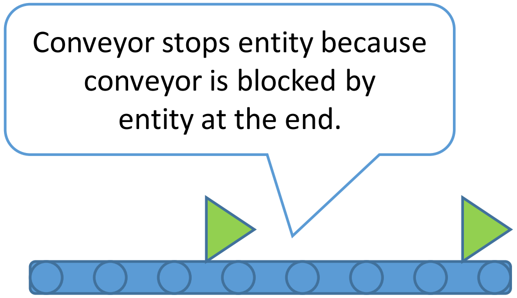
Conveyor – May be used to move Entities between two locations using an accumulating or non-accumulating Conveyor.

- Accumulating Type: Allows accumulation of Entities at the end of the Conveyor if there is blockage.
- Non-Accumulating Type: If there is blockage at the end of the Conveyor then all movement on the Conveyor is stopped.

Tip A Conveyor can optionally be configured with fixed interval spacing. For example, to model a power & free conveyor with ‘dogs’ hanging at set intervals.

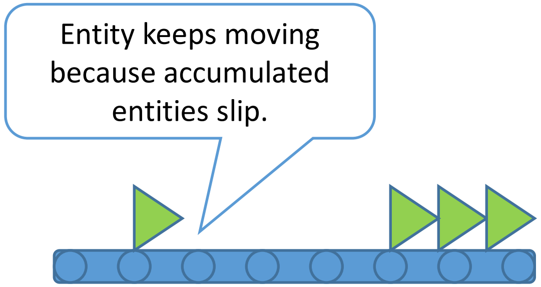
Conveyor

- Entity movement is controlled by the Conveyor.
- A Conveyor stops when blocked and then restarts using *Desired Speed*.
- *Accumulating* option allows slippage for blocked Entities and continued Conveyor movement for non-blocked Entities.
- *Entity Alignment* can be at 'Any Location' or at equally spaced 'Cell Locations'.
- Also incorporates Reliability, Travel Capacity, Drawn to Scale option, and Selection Weight.



Conveyor stops entity because conveyor is blocked by entity at the end.

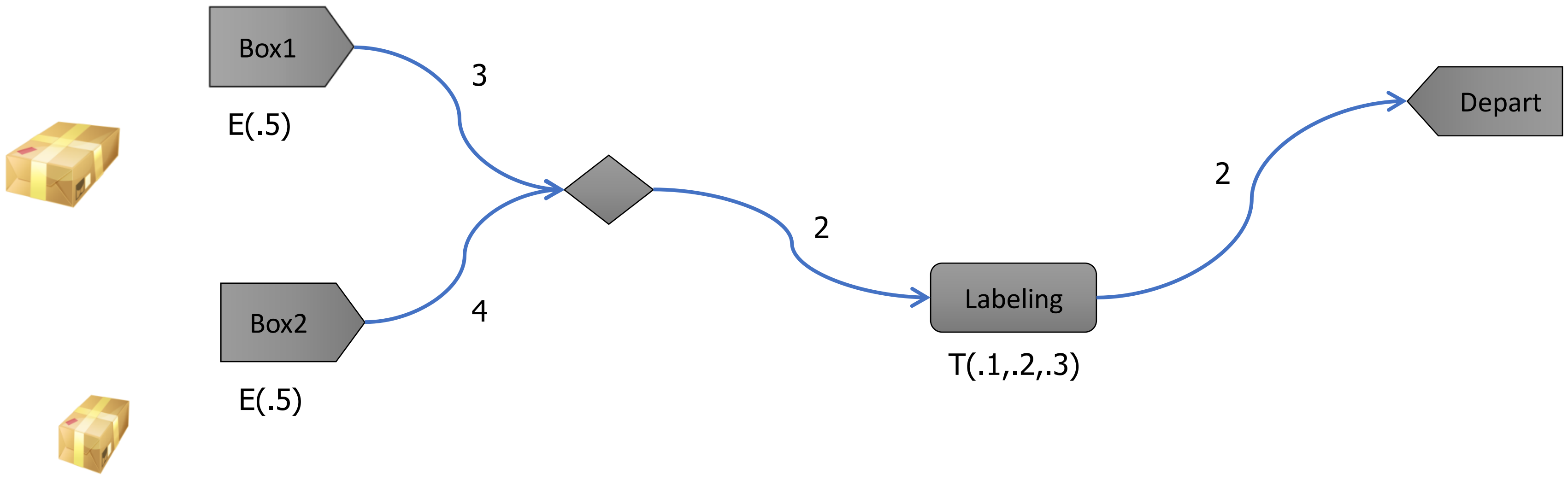
Accumulating = False



Entity keeps moving because accumulated entities slip.

Accumulating = True

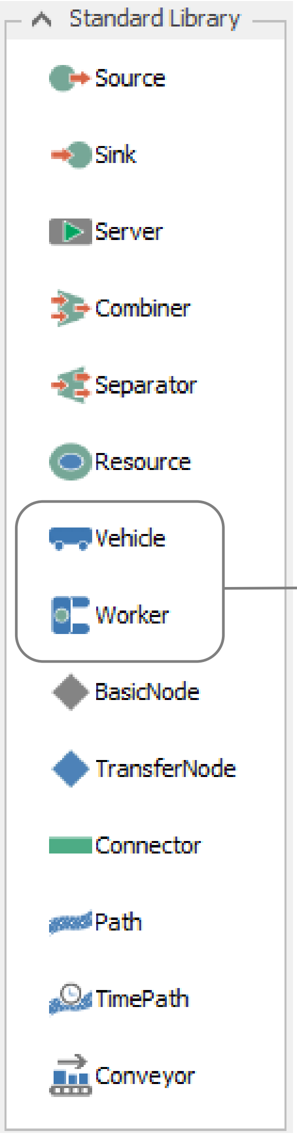
Example: Merging Conveyors



Material Handling

Material Handling Using Vehicles and Workers

Material Handling using Workers/Vehicles

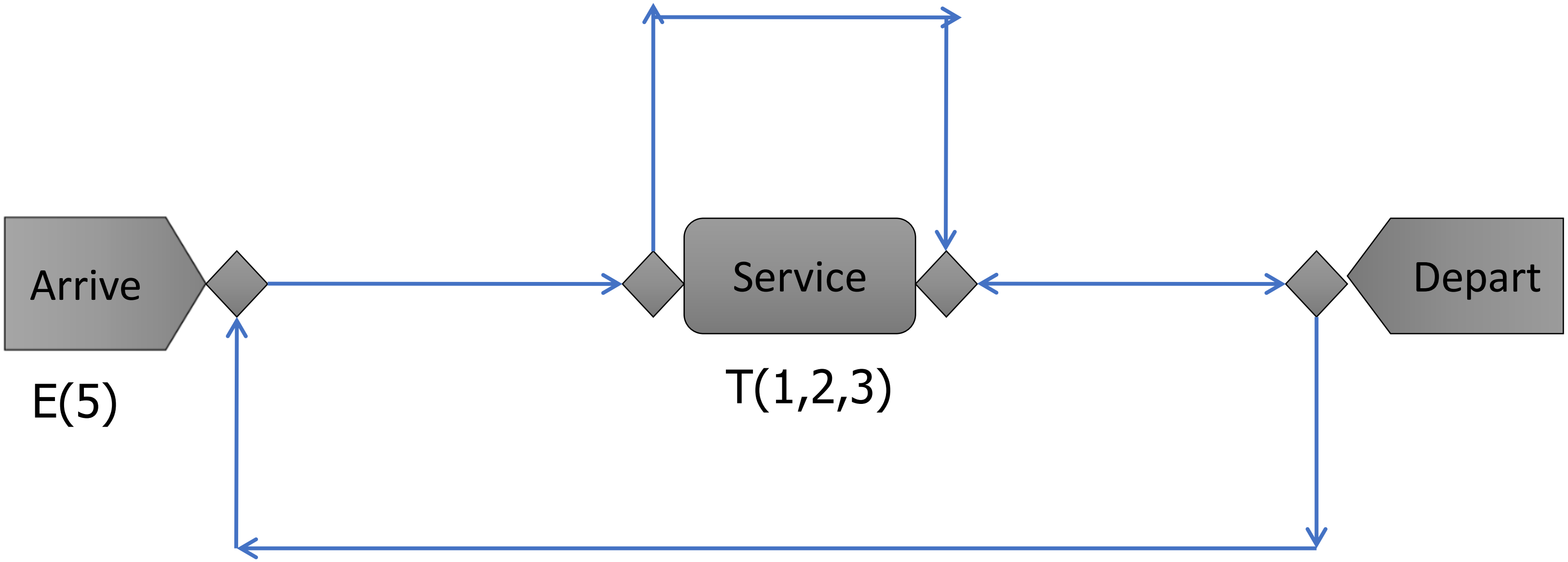


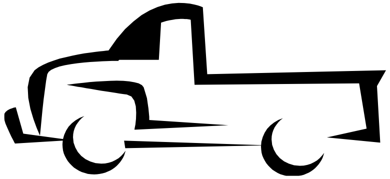
Vehicle – May be used to model a device for loading, transporting, and unloading of other Entities.

Worker – May be used to model a labor resource whose tasks include loading, transporting, and unloading of other Entities.

Tip ☞ All of Simio's standard library objects provide 'Add-On Process' support. An *Add-On Process* is an additional, user-defined process routine specified to be executed whenever a particular type of event related to the object occurs (e.g., whenever a vehicle or worker is evaluating a transport request, is loading, or is unloading, etc.).

Example: On-Demand Pickups

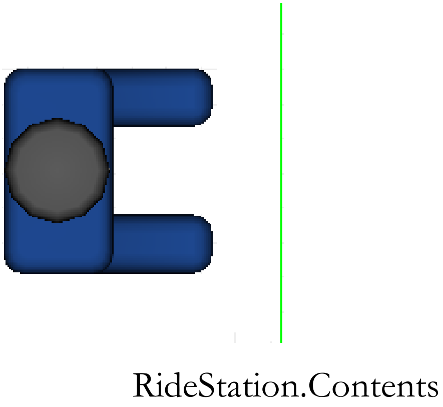




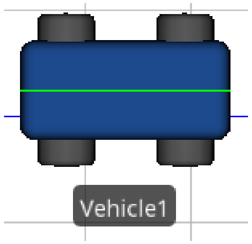
0.3 min load/unload
Home: Input@Depart

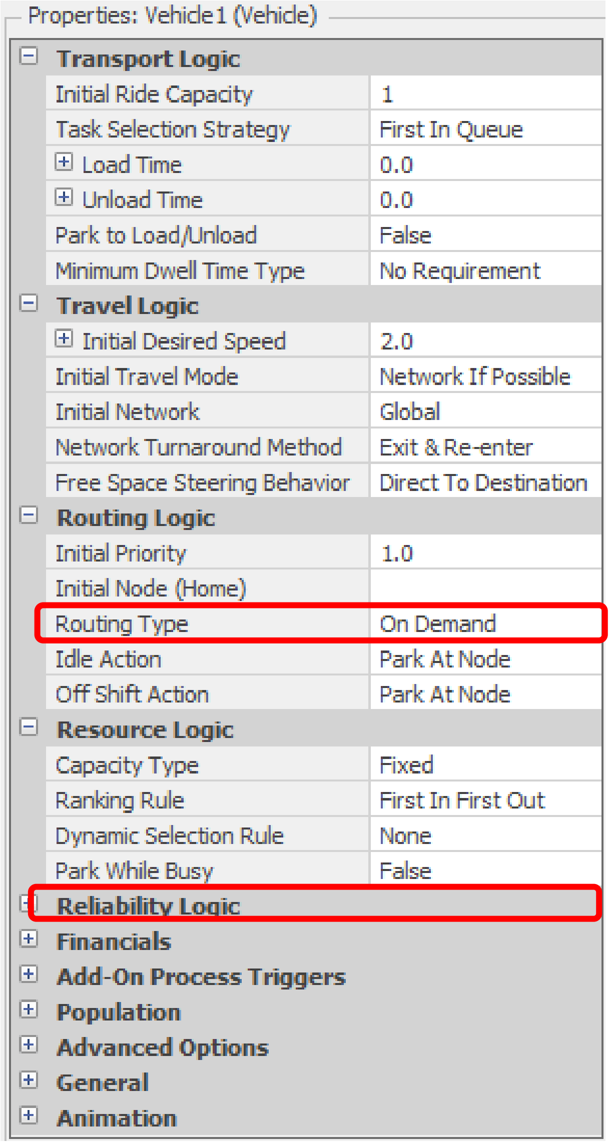
Worker

- Worker can perform stationary tasks, when Seized / Released – it can move (optionally) to Entity location first, then perform task.
- Worker can also pickup, carry, and drop-off Entities, like Vehicle.
- Worker can have Fixed capacity or Work Schedule.
- A Worker has a Desired Speed, Ride Capacity, and Load/Unload Times.
- Worker has Evaluating Transport Request and Evaluating Seize Request logic to accept or reject various tasks.



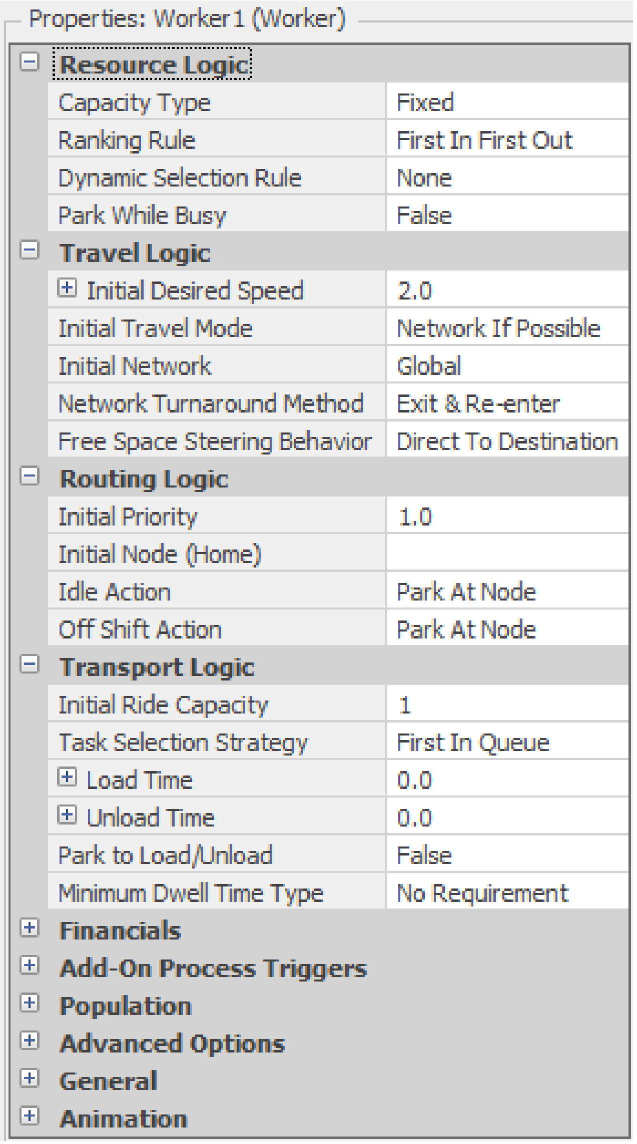
Vehicle vs Worker





Properties: Vehicle1 (Vehicle)

⊟ **Transport Logic**	
Initial Ride Capacity	1
Task Selection Strategy	First In Queue
⊞ Load Time	0.0
⊞ Unload Time	0.0
Park to Load/Unload	False
Minimum Dwell Time Type	No Requirement
⊟ **Travel Logic**	
⊞ Initial Desired Speed	2.0
Initial Travel Mode	Network If Possible
Initial Network	Global
Network Turnaround Method	Exit & Re-enter
Free Space Steering Behavior	Direct To Destination
⊟ **Routing Logic**	
Initial Priority	1.0
Initial Node (Home)	
Routing Type	On Demand
Idle Action	Park At Node
Off Shift Action	Park At Node
⊟ **Resource Logic**	
Capacity Type	Fixed
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Park While Busy	False
⊞ **Reliability Logic**	
⊞ **Financials**	
⊞ **Add-On Process Triggers**	
⊞ **Population**	
⊞ **Advanced Options**	
⊞ **General**	
⊞ **Animation**	



Properties: Worker1 (Worker)

⊟ **Resource Logic**	
Capacity Type	Fixed
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Park While Busy	False
⊟ **Travel Logic**	
⊞ Initial Desired Speed	2.0
Initial Travel Mode	Network If Possible
Initial Network	Global
Network Turnaround Method	Exit & Re-enter
Free Space Steering Behavior	Direct To Destination
⊟ **Routing Logic**	
Initial Priority	1.0
Initial Node (Home)	
Idle Action	Park At Node
Off Shift Action	Park At Node
⊟ **Transport Logic**	
Initial Ride Capacity	1
Task Selection Strategy	First In Queue
⊞ Load Time	0.0
⊞ Unload Time	0.0
Park to Load/Unload	False
Minimum Dwell Time Type	No Requirement
⊞ **Financials**	
⊞ **Add-On Process Triggers**	
⊞ **Population**	
⊞ **Advanced Options**	
⊞ **General**	
⊞ **Animation**