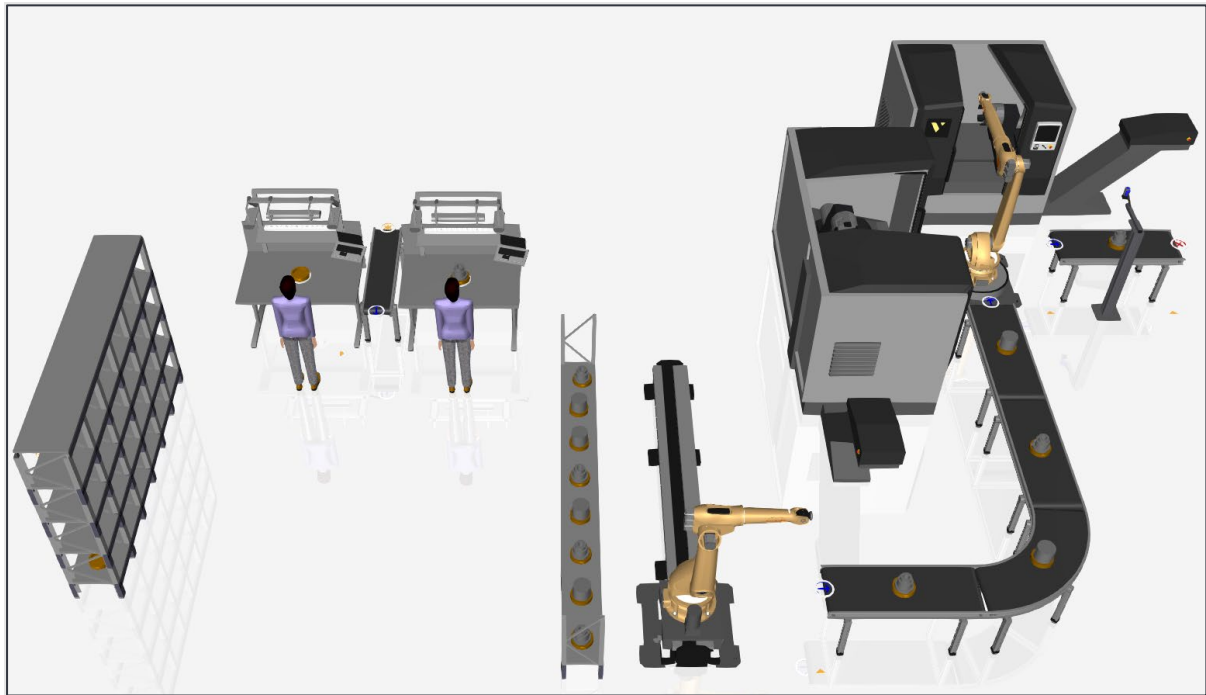


Warehousing components reference

Visual Components 4.4 | Version: February 08, 2022



This living document introduces warehouse components in the eCatalog, their properties, and simple use cases. It will be updated as more warehouse-related components are added or updated in the eCatalog.

NOTE! There are separate tutorials for resources commonly used with warehouse components, such as the [ASRS manual](#) introducing cranes.

You will go through properties and use cases of the following:

- Warehouse Shelf / Floorspace Buffer
- Warehouse Process Shelf
- Dual Rail Transport Controller
- Overhead Cranes
- Stack Feeder
- Pallet Feeder

Support

support@visualcomponents.com

Visual Components Forum

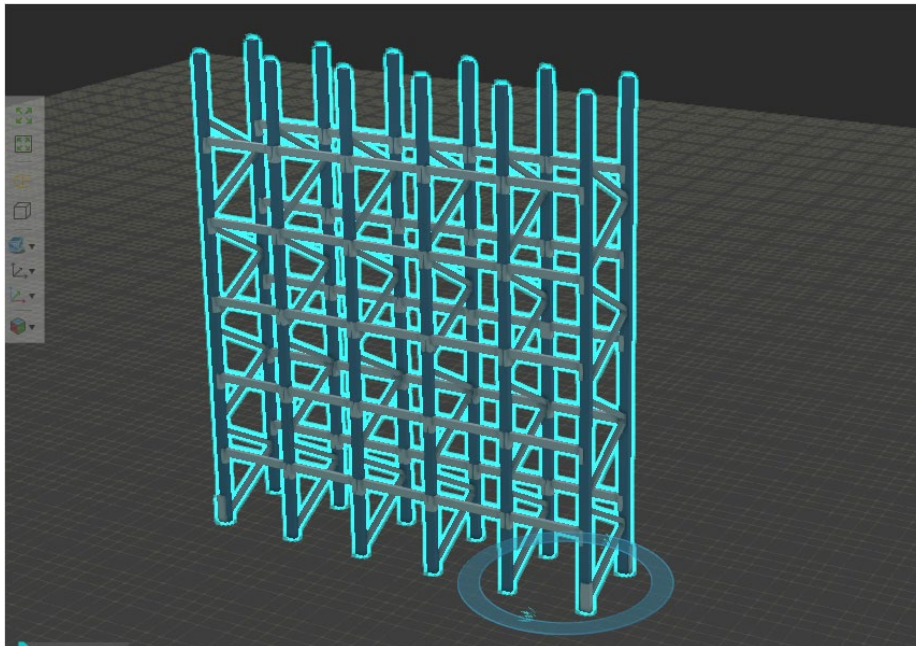
forum.visualcomponents.com

Contents

Warehouse Shelf / Floorspace Buffer.....	3
Warehouse Process Shelf.....	7
Properties.....	8
Transport Link properties.....	9
Stack Feeder.....	12
Pallet Feeder	14

Warehouse Shelf / Floorspace Buffer

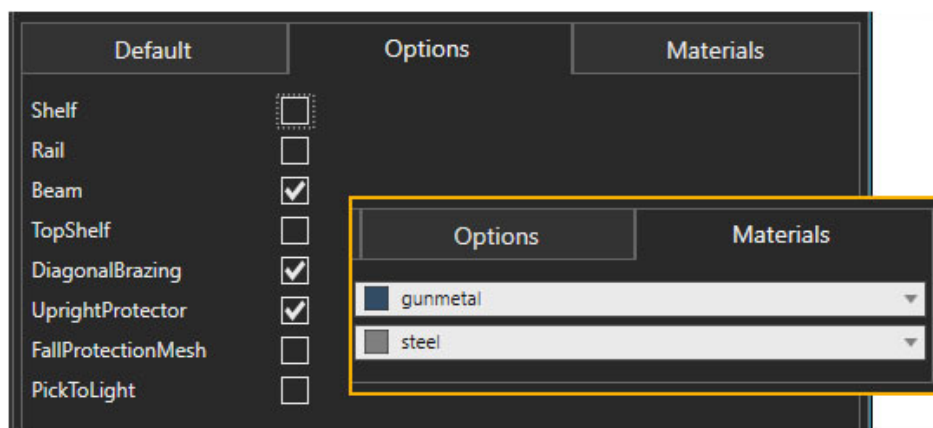
A Warehouse shelf is a Process Modelling (PM) process for storing and buffering products.



The warehouse includes multiple presets that can be used to set up the appearance and dimensions of the following:

- Pallet Rack
- Crane Rack
- Light Pallet Rack
- Manual Picking Shelf
- PickToLight

The Visibility of different visual features can be manually changed from the **Options** tab. In addition to a component's basic properties, you can change the color of different shelf parts in the **Materials** tab.



Single shelf bay dimensions, **Tiers** and **Bays**, and other structural properties can be changed from the component's basic properties.

Component Properties

Warehouse Shelf

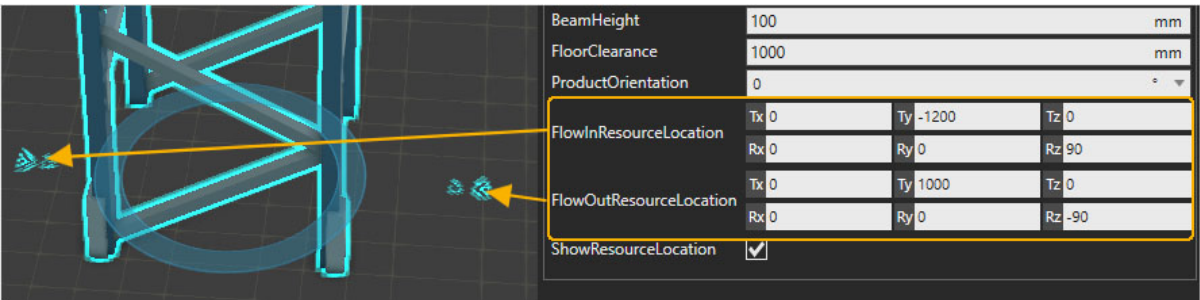
Tiers	5
Bays	5
BaysPerSupportBeam	1
BayWidth	900 mm
BayDepth	1100 mm
BayHeight	1000 mm
BayTilt	0

You can use the **ProductOrientation** property to define the orientation of the product while on a shelf.

FloorClearance	1000	mm
ProductOrientation	0	°
FlowInResourceLocation	Tx 0 Ty -1200 Tz 0	Rx 0 Ry 0 Rz 90
FlowOutResourceLocation	Tx 0 Ty 1000 Tz 0	Rx 0 Ry 0 Rz -90
ShowResourceLocation	<input checked="" type="checkbox"/>	

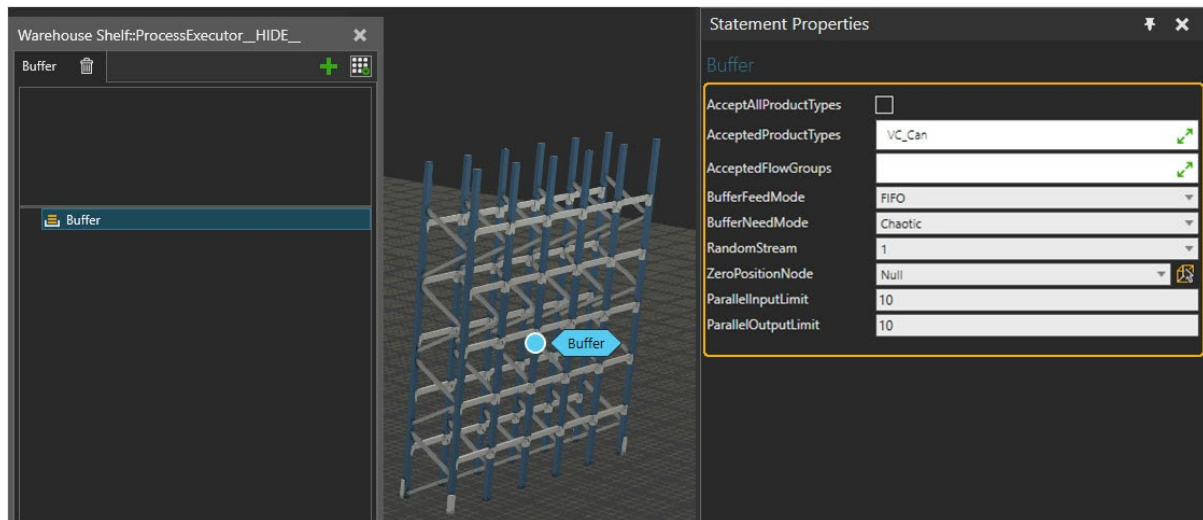
It is possible to load and unload the shelf from opposite sides. If resources such as humans are used, they will go to the **FlowIn** or **FlowOutResourceLocation**. Cranes or robot arms will not use the location.

A small arrow indicates the flow in or out of the location. The location will automatically transform to match the product location on the floor level, and the offset from the shelf will be maintained. Both locations can be modified with the corresponding properties on the shelf.

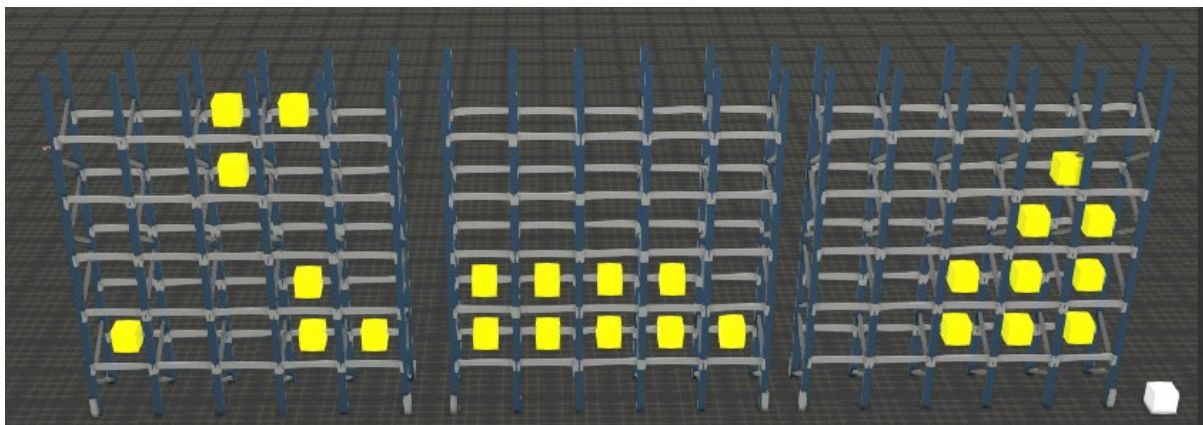


Some statement properties need to be visited to change, for example, the following:

- **AcceptAllProductTypes**
- **AcceptedProductTypes**
- **AcceptedFlowGroups**
- **BufferFeedMode**
- **BufferNeedMode**
- **ParallelInputLimit**
- **ParallelOutputLimit**

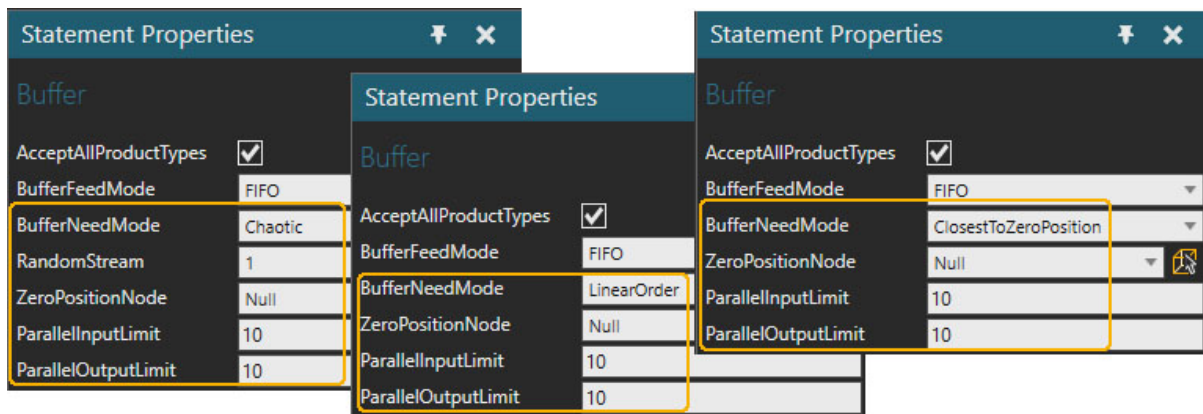


The **ParallelInputLimit** should be set to a value less or equal to the total capacity of the available resources. That way, if more than one shelf is used, the shelves will be filled roughly at the same rate. **ParallelOutputLimit** is set automatically to match the number of slots but can be manually changed after the shelf has been set up.

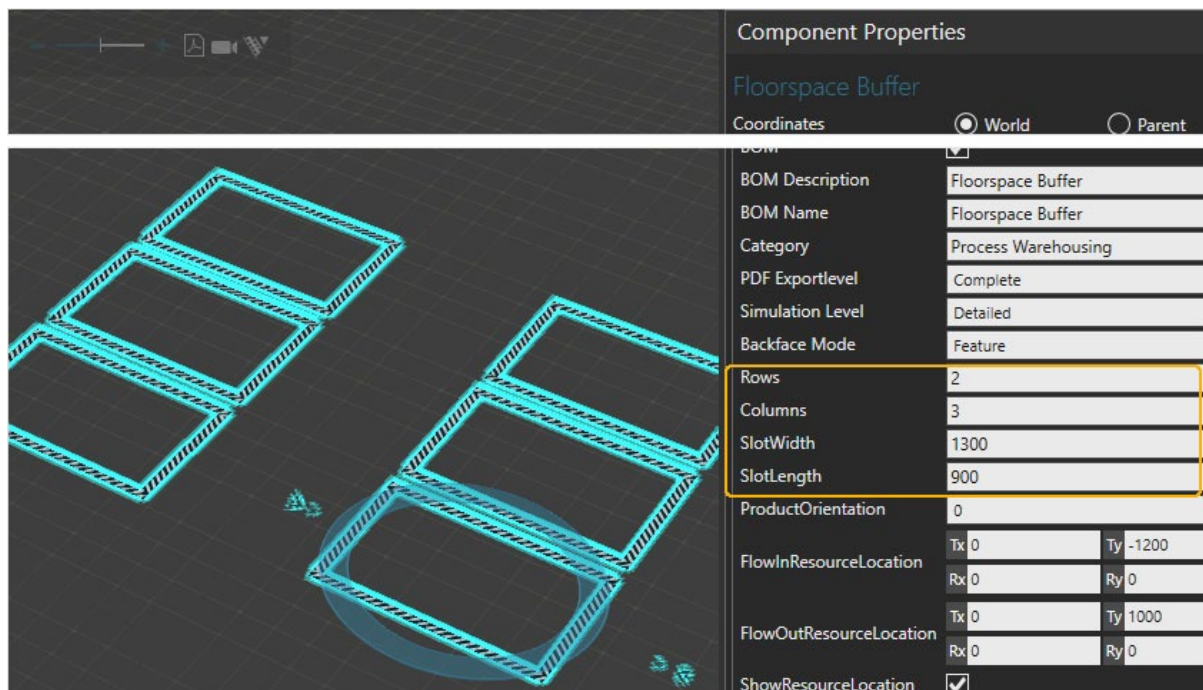


Examples from the **BufferNeedMode** while the block component is selected as **ZeroPositionNode**:

- Left is *Chaotic*
- Middle is *LinearOrder*
- Right is *ClosestToZeroPosition*



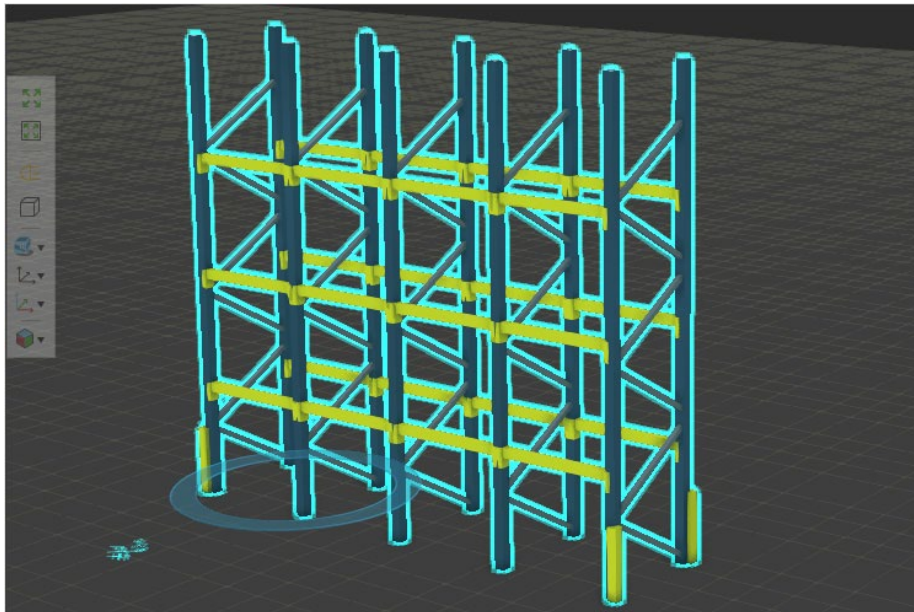
The Floorspace buffer component shares the functionality of the Warehouse Shelf component. Instead of the shelf structure, the Floorspace Buffer component utilizes floor space as a buffer. Single-slot dimensions, spacing, and the number of **Rows** and **Columns** can be changed from the **Component Properties**.



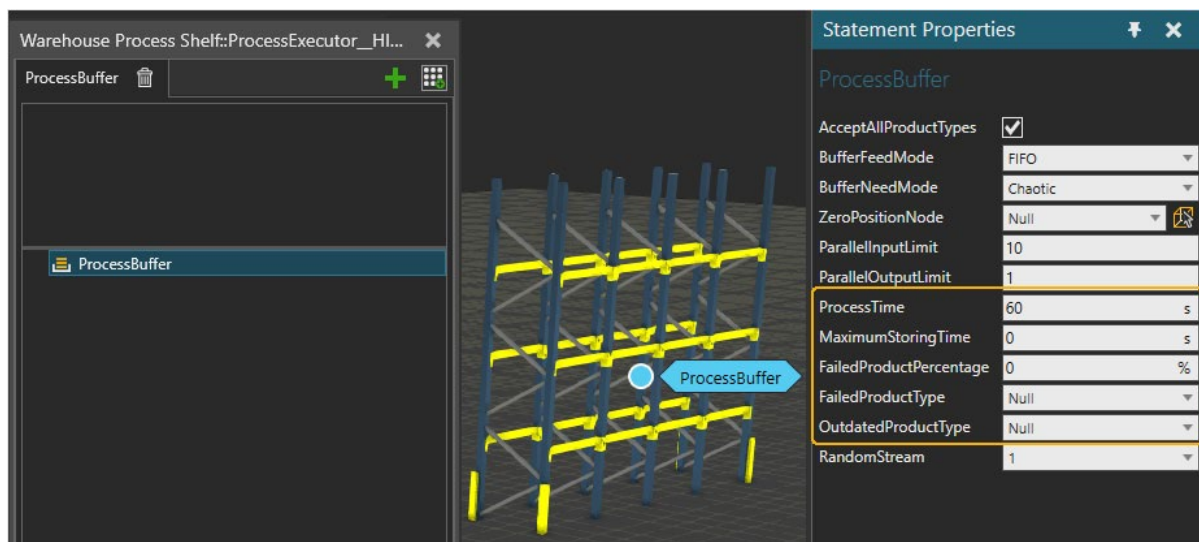
Warehouse Process Shelf

A Warehouse Process Shelf is a Process Modelling (PM) process for curing and processing multiple products simultaneously. Typical applications are curing shelves, test rack processes, and monitoring or limiting a products' storing time.

Components share the same properties with the Warehouse Shelf in the **Component Properties** tab.



A Warehouse process shelf has some additional properties in **Statement Properties**.



ProcessTime can set the processing time for each product placed on the shelf. After this time is consumed, the product is ready to leave the process.

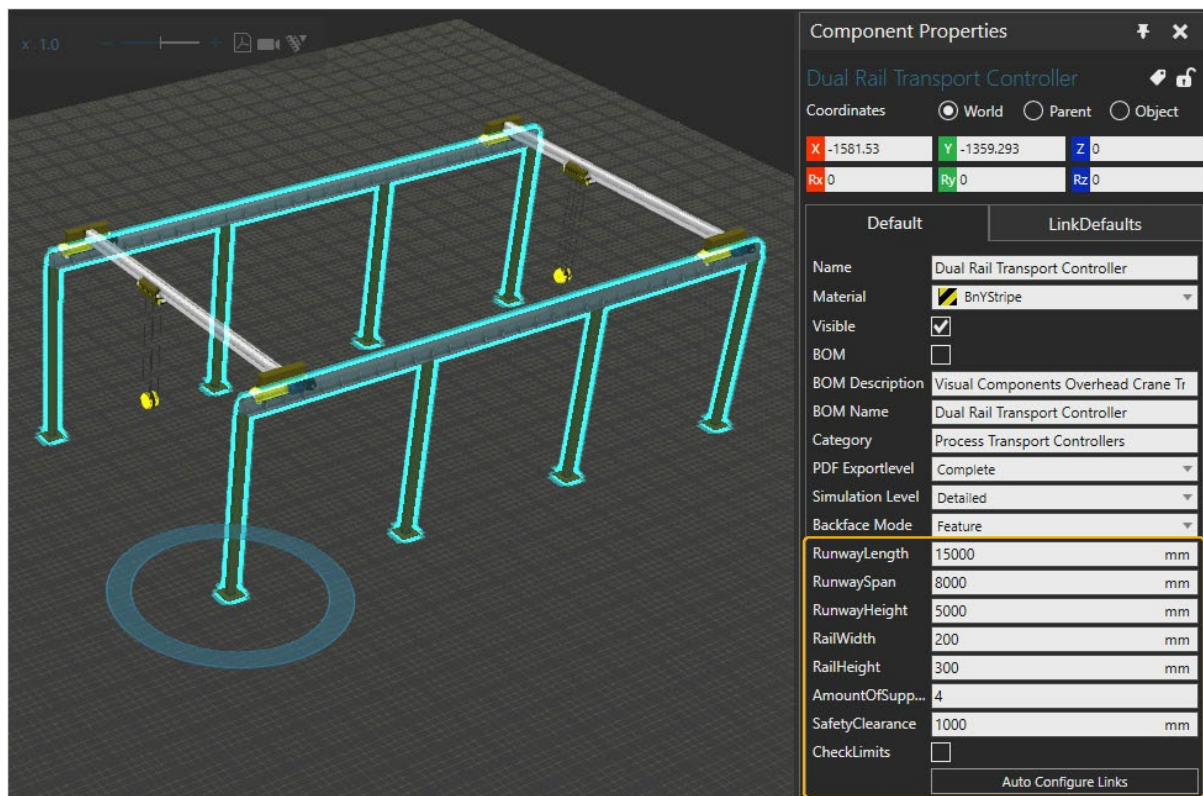
If **MaximumStoringTime** is defined (other than zero) and the product remains on the shelf longer than defined in this property (in addition to possible **ProcessTime**), the product will become *Outdated*, and its product type will be changed to **OutdatedProductType**.

If **FailedProductPercentage** is defined, this value is used as a probability that a product will become defective after processing, and its product type is changed to **FailedProductType**.

Dual Rail Transport Controller

The Dual Rail Transport Controller is required to control the cranes and serve as the track/runway for the cranes. It can control up to two cranes simultaneously. The cranes are connected to the controller using the **PnP** tool. The cranes are referred to as crane A or B, determined by the interface the crane is connected to.

Once a crane is connected, the interface matches the crane's dimensions, preventing it from exceeding the track. The dimensions of the crane are taken from its bounding box.



Dual Rail Transport Controller with two overhead cranes, Crane A on the left and Crane B on the right

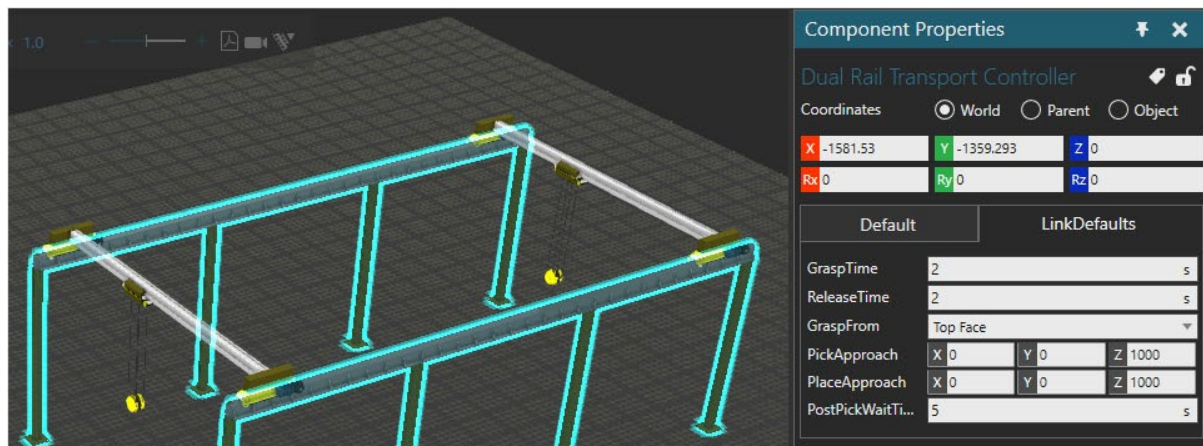
Properties

RunwayLength, RunwaySpan, and RunwayHeight: The dimensions of the track

SafetyClearance: Determines the minimum separation between the cranes if more than one crane is connected to the same track

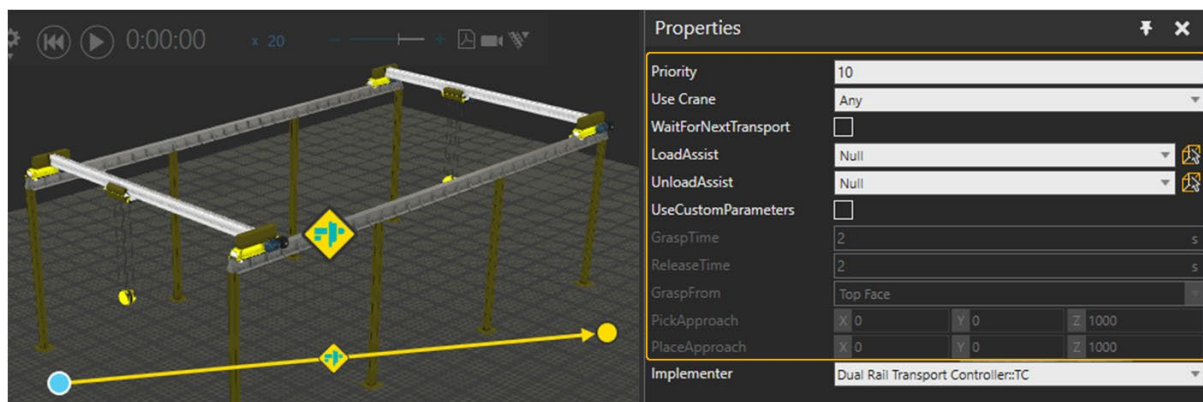
CheckLimits: Pause the simulation and display output message if a crane runs out of track (exceeds the track length)

Auto Configure Links: If more than one crane is connected to the track, auto-configure will associate each link with crane A or B using the crane closest to the link's source and destination nodes.



When the Dual Rail Transport Controller is assigned to a transport link, some default properties and their values in the **LinkDefaults** tab in its **Component Properties** panel are applied to transport link(s). If the values in the **LinkDefaults** tab are changed the matching values in the transport links will also change, if UseCustomParameters is not enabled in the transport link.

Transport Link properties



Priority: Sets the priority for the link against all available/active transportation tasks. The lower the number, the higher the priority

Use Crane: Define a specific crane for transportation or allow the controller to choose and crane. The controller will, by default, select the nearest available.

WaitForNextTransport: If enabled, a crane will wait and remain reserved until the process executions reach the next *WaitTransport* statement in the process. This process is used in conjunction with the *StartTransportIn* statement.

LoadAssist, UnloadAssist: Controller of a resource that will move into the loading/unloading position of the crane. The process is executed in this position (**GraspTime/ReleaseTime**).

UseCustomParameters: Enable to override default values (defined in controller properties)

GraspTime: A time that is taken to load the crane. Excluding handler extend and retract times.

ReleaseTime: A time that is taken to unload the crane. Excluding handler extend and retract times.

GraspFrom: Selection from which position a product is transported.

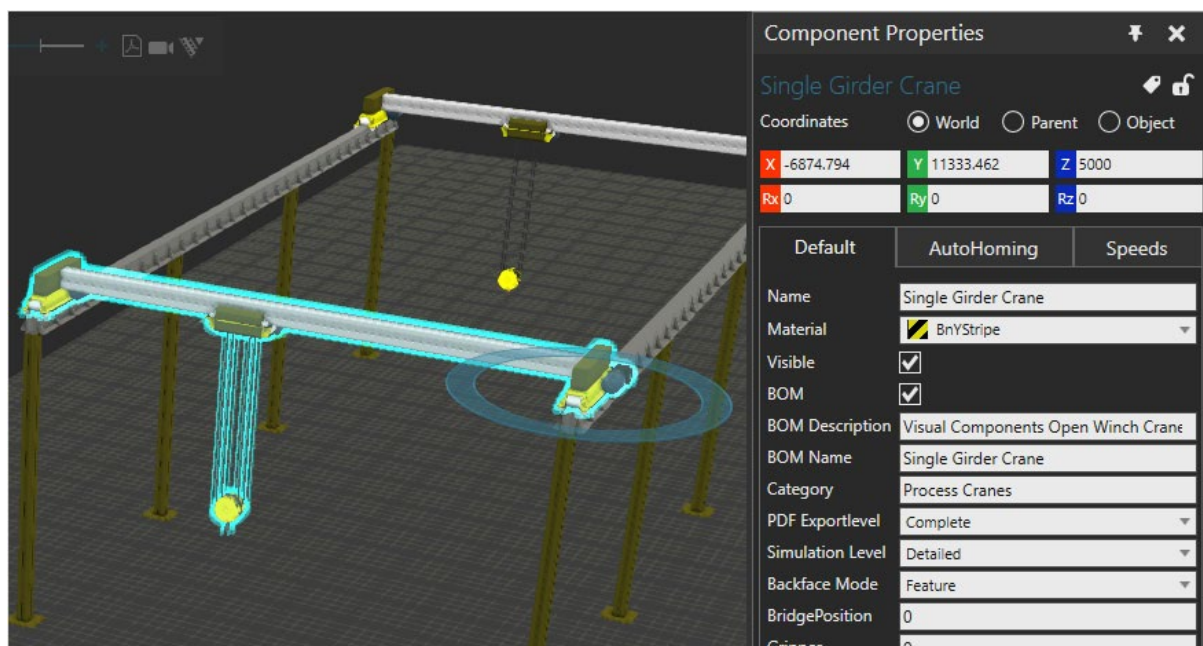
PickApproach, PlaceApproach: Linear approach direction and distance of the crane when the product is grasped/released.

Overhead Cranes

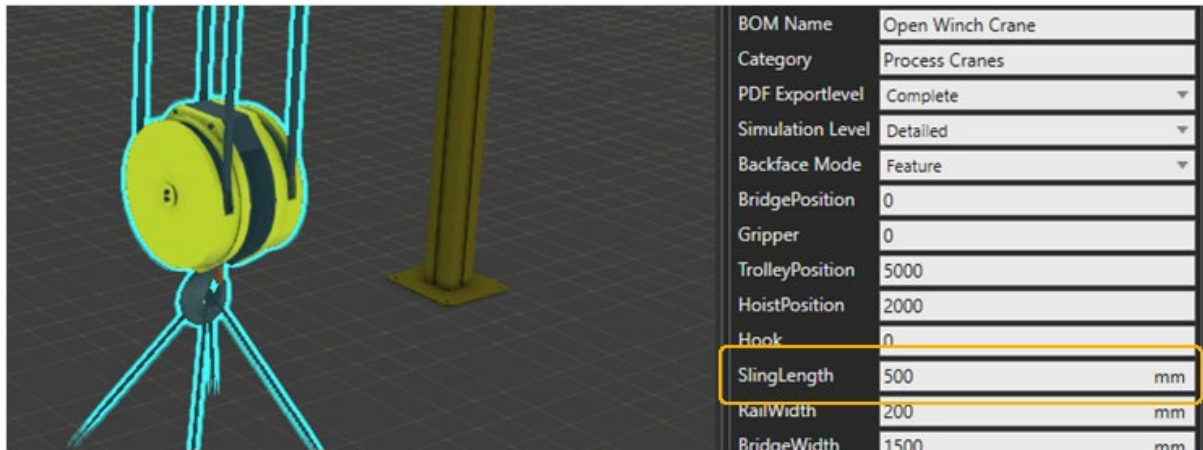
Several process modeling cranes are available in the **eCatalog - PM Cranes**. Cranes vary in construction and size, but their behaviours and properties are the same.



Up to two overhead cranes can be connected to a Dual Rail Transport Controller



Cranes have a set of default properties (like **RailWidth**) to define the size and robustness of different parts of the crane.



SlingLength can be used to define additional distance from the crane hook to the product and visualize lifting slings.

AutoHoming::Delay: Defines the time once the crane has been without tasks to move to a home position defined by the following properties. Use zero to disable **AutoHoming**.

AutoHoming::BridgePosition: Bridge (X) home position

AutoHoming::TrolleyPosition: Trolley (Y) home position

AutoHoming::HoistPosition: Hoist (Z) home position

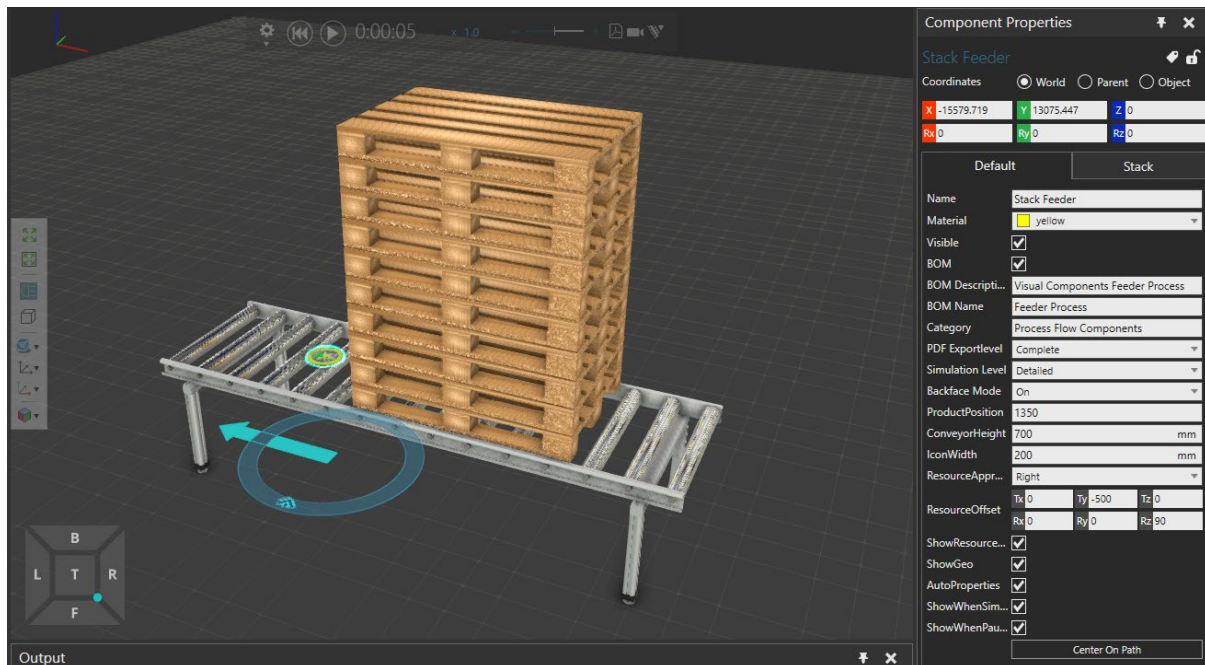
AutoHoming::Read Current Joint Values: Read the current position and set the above properties.

Default	AutoHoming	Speeds
Delay	5 s	
BridgePosition	0 mm	
TrolleyPosition	5000	
HoistPosition	2000 mm	
Read Current Joint Values		

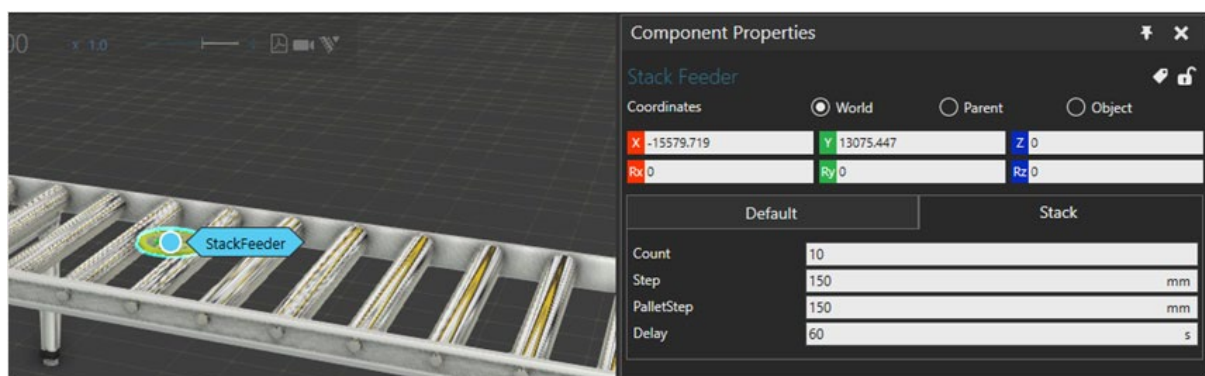
The **Speeds** tab properties set the kinematic parameters for each joint. The deceleration will be equivalent to the acceleration.

Stack Feeder

A Stack feeder is a component for creating stacks of product instances in Process Modelling. The bottom product in a stack may be different from the rest of the stack. The component uses a Process Modelling routine controlled by **Component Properties**.



A Stack feeder is a process that can be connected to a conveyor. It shares default properties with all Process Modelling flow components.



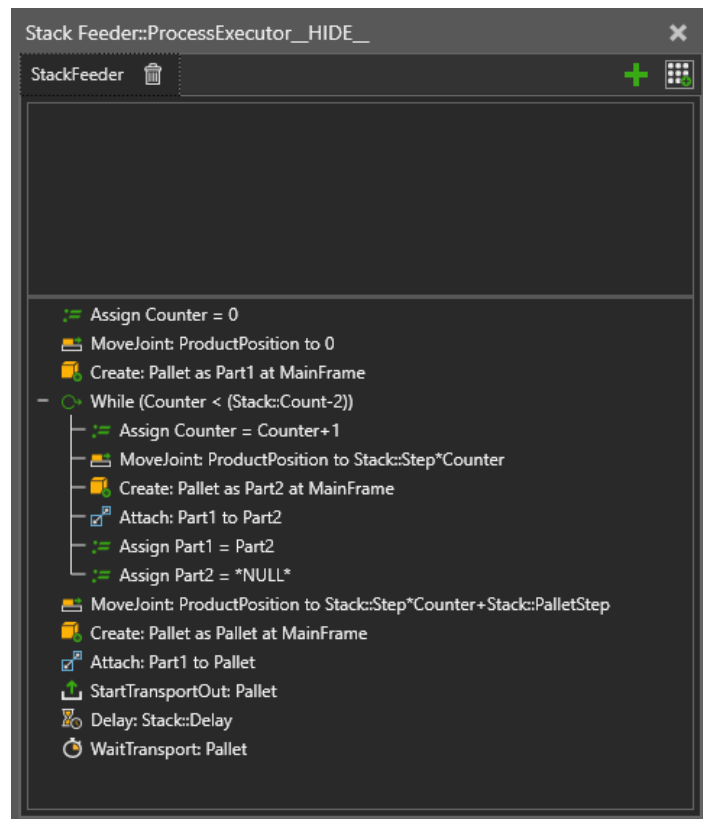
Properties in the **Stack** tab control the *StackFeeder* routine.

In this routine, a Joint modeled to a component is moved up while **Stack::Count** monitors the number of products created. Created products are attached, creating a hierarchical stack. The distance between products in the stack is defined with the **Stack::Step** property.

The Last product in the stack is created separately, and the distance to the stack is defined with separate **Stack::PalletStep** property. This allows you to select a different product type as the bottom part of the stack.

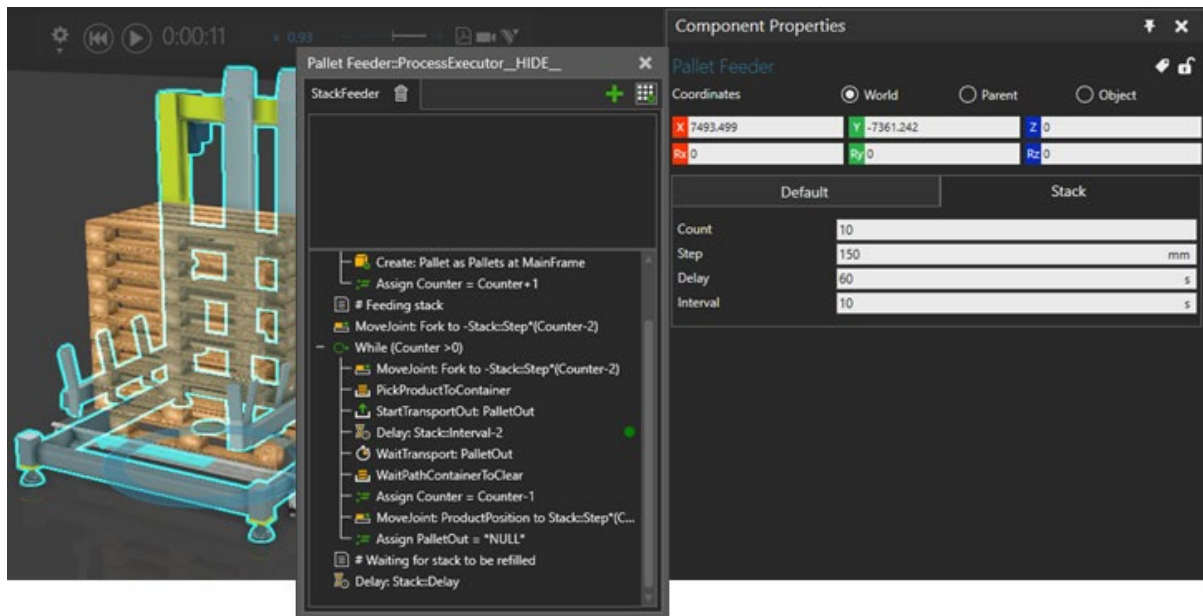
After a stack is created, it is transported out from the process and is delayed the amount of time in **Stack::Delay**.

The Stack feeder creates stacks using a pallet as its default product type. Stacks are created using Create statements and the product type can then be modified manually once a new product type is selected in the product type editor.



Pallet Feeder

A Pallet feeder is a component for creating stacks of product instances in Process Modelling. This component uses a Process Modelling routine controlled by **Component Properties**, and routine execution is partly controlled by Python Statements.



StackFeeder is a process that can be connected to a conveyor. It shares default properties with all Process Modelling flow components.

The feeder has a set of default properties (like **ProductLength**) to define the size and visibility of different parts of the feeder.

Properties in the **Stack** tab control the *StackFeeder* routine.

In the routine, a Joint modeled to a component is moved up while **Stack::Count** controls the number of products created. The **Stack::Step** property defines the distance between products in a stack.

After a stack is created, products are transported out from the process one by one with **Stack::Interval**, and then the whole stack is transported out, **Stack::Delay** controls the amount of time the products are delayed.

The Stack feeder creates stacks using a pallet as its default product type. Stacks are created using Create statements and the product type can then be modified manually once a new product type is selected in the product type editor.