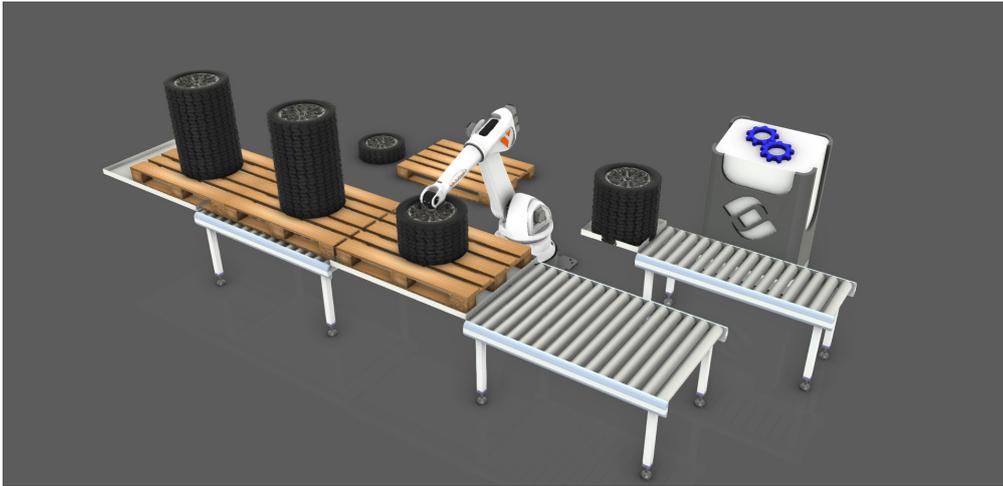


# Works Library - Basics of Task Creation

Visual Components 4.0 | Version: 9.2.2017



The Works library can be used to simulate simple and complex processes during a simulation. A process contains tasks, which can be self-contained or require the use of resources, machines, inputs and outputs, and other processes.

In this tutorial you will learn the basics of creating and simulating tasks with components in the Works library.

This will involve:

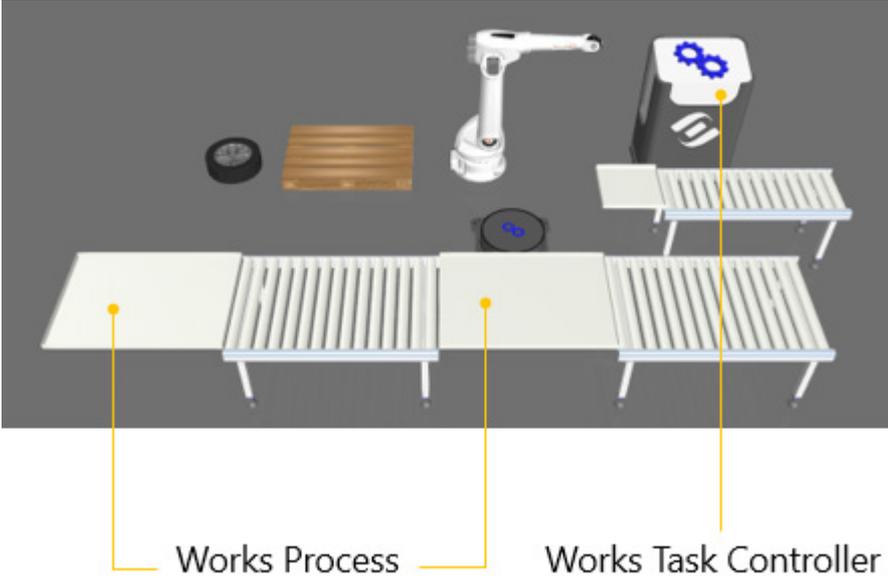
- Creating products and forming stacks
- Teaching locations for storing products and filling containers
- Transporting products in and out of processes
- Using robot resources to pick and place parts

**Support**  
[support@visualcomponents.com](mailto:support@visualcomponents.com)

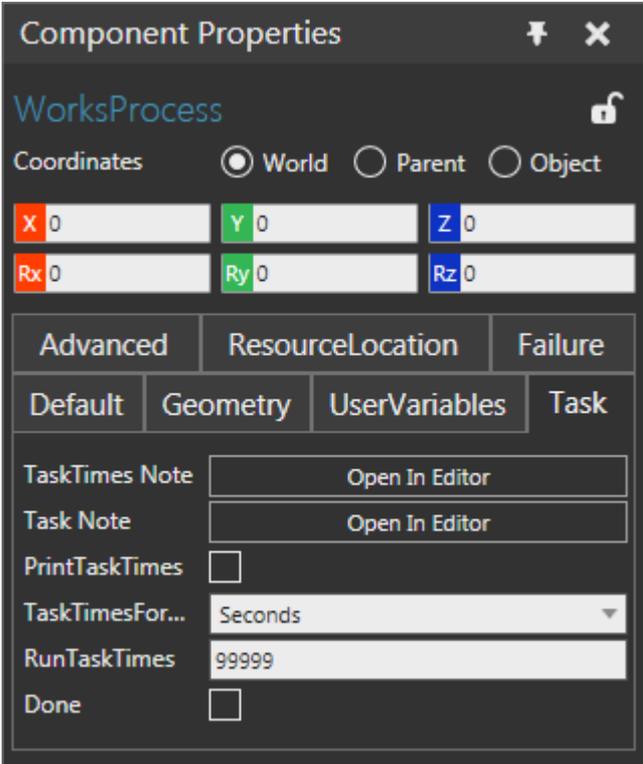
**Community**  
[community.visualcomponents.net](http://community.visualcomponents.net)

# Getting Started

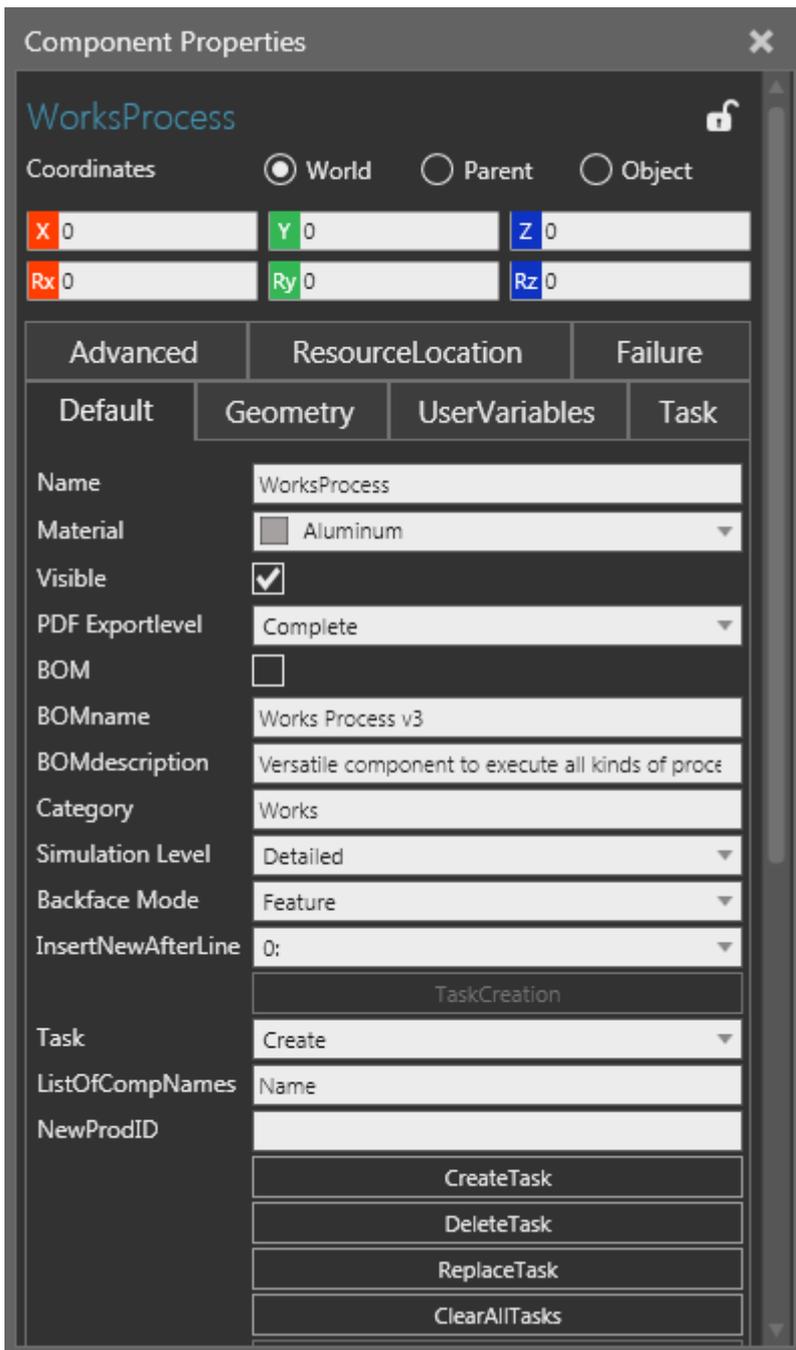
A Works Process component is used to create and execute tasks during a simulation. A Works Task Controller is used to manage the execution of those tasks. As a result, a layout with a Works Process component requires at least one Works Task Controller.



By default, a Works Process component executes tasks in sequence. This process can include conditional tasks, calls to global processes in the Works Task Controller, and be executed multiple times during a simulation. The work order or tasks performed by a Works Process component can be found in its Task properties group.



You can add, edit, delete and replace tasks in a Works Process component by using its Default properties group.

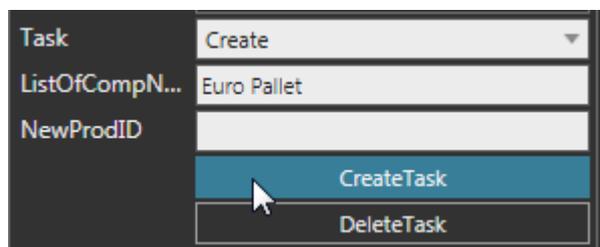


**NOTE!** For advanced users, you can access and edit the Task note of a Works Process component. We recommend learning and understanding the syntax of tasks before directly editing a Task note.

## Task - Create

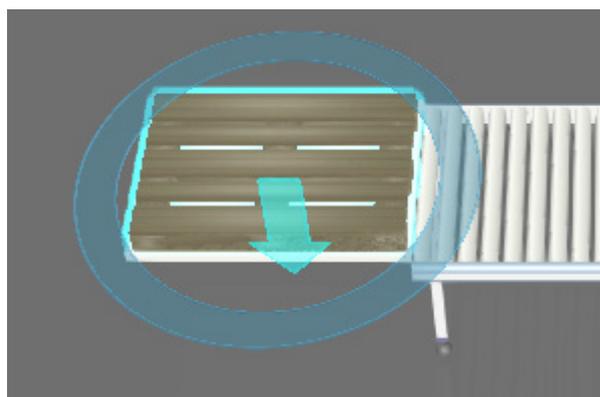
A Create task allows you to create a list of components found in the 3D world. A component needs to be referenced by name. At the time of creation, the new component can be assigned a specific ProdID. Otherwise, the default ProdID of a created component is its name.

1. Press CTRL+N to clear the 3D world.
2. Open the **Task Creation - Start** layout for this tutorial.
3. In the Cell Graph panel, under the Works category, click **WorksProcess** to select that component in the 3D world.
4. In the Component Properties panel, Default tab, set Task to **Create** and ListOfCompNames to **Euro Pallet**, and then click **CreateTask**.

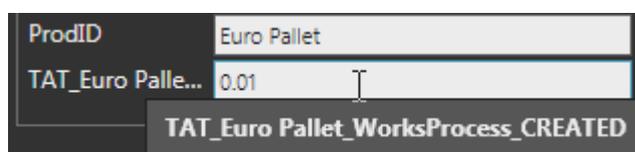


**TIP!** Use the Cell Graph panel to quickly reference the names of components in the current layout. The InsertNewAfterLine property can be used to quickly verify task creation was successful and the position of the new task in the process.

5. Run the simulation to verify a pallet is created at the default frame location in the Works Process component, and then stop the simulation.
6. Select the **new pallet**. Notice it is attached/contained in the Works Process component.



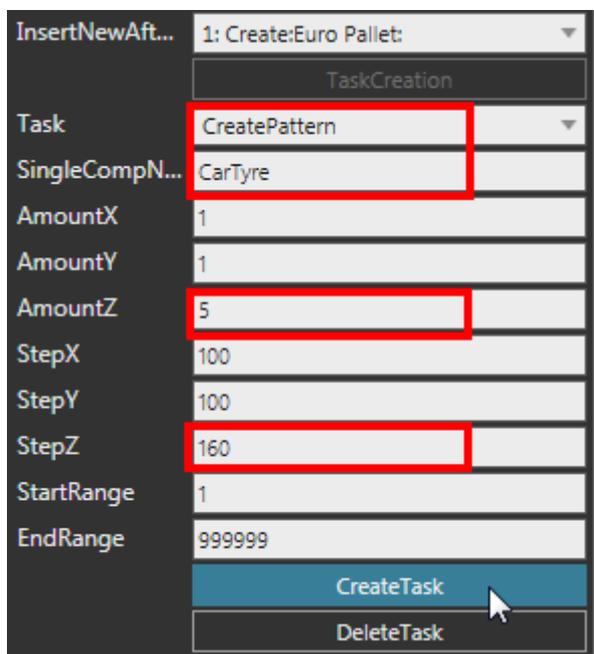
7. In the Component Properties panel, verify the **ProdID** of the pallet is its name and notice it has a stamped **TAT\_** property to indicate its turnaround time in the process.



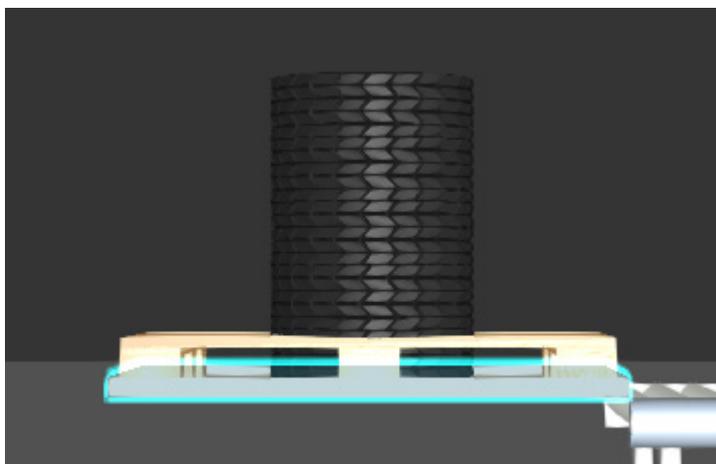
# Task - Create Pattern

A Create Pattern task allows you to create a pattern of components by using a template. The template can be any component found in the 3D world. AmountX, Y and Z properties define the size of the pattern along the XYZ axes. StepX,Y and Z properties define the spacing of the pattern along the XYZ axes. StartRange and EndRange properties define what elements in the pattern are generated by the task.

1. Reset the simulation, and then select the **WorksProcess** component.
2. In the Component Properties panel, Default tab, set Task to **CreatePattern**, SingleCompName to **CarTyre**, AmountX and Y to **1**, AmountZ to **5** and StepZ to **160**, and then click **CreateTask**. Notice that you are inserting a new task after you create a pallet in the process.



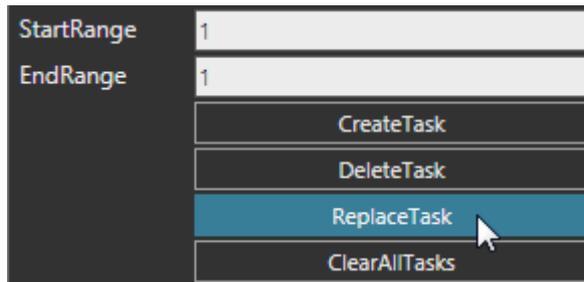
3. Run the simulation to verify a stack of tires are created, and then stop the simulation. Notice that the location of the first tire in the pattern is inside the pallet. You can fix this by teaching the Works Process component a default location for the tire.



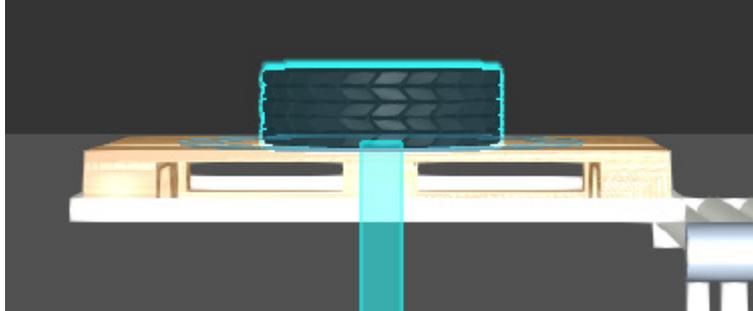
# Teach Locations

Locations allow you to define where you want components to be placed, stored and moved to in a Works Process component. For example, you can teach the location for the first item in a pattern.

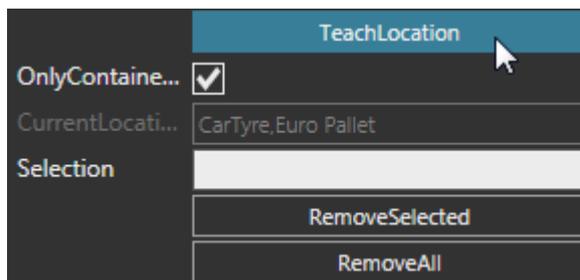
1. Reset the simulation, and then in the Component Properties panel, Default tab, set EndRange to **1**, and then click **ReplaceTask**. This will replace the task listed in the InsertNewAfterLine property, which in this case is the CreatePattern task for the tires. Now, the pattern will only create one item, thereby making it easier to teach the location of the tire to the Works Process.



2. Run the simulation, verify there is one tire inside the pallet, and then stop the simulation.
3. Select the **CarTyre** inside the pallet, and then in the Component Properties panel, set the Z-axis coordinate to **844**. This offsets the tire to be on top of the pallet.



4. In the 3D world, select the **WorksProcess**, and then in the Component Properties panel, Default tab, click **TeachLocation**. The WorksProcess only checked the locations of components attached to it because the OnlyContainedComponents check box is enabled. This is why there are saved locations for a CarTyre and Euro Pallet.

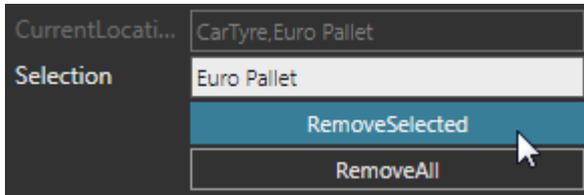


**NOTE!** A location is a key-value pair. The *key* is referring to either the name or ProdID of a component. The *value* is the location of a component based in the Works Process coordinate system. ProdID values can be used to teach different locations for the same type of component.

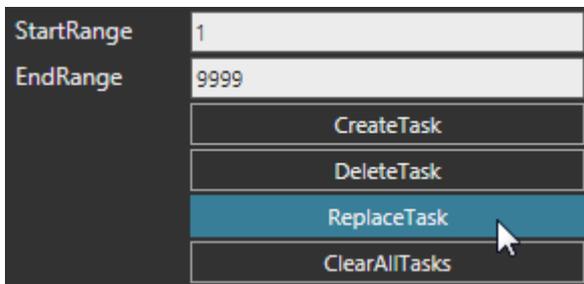
# Remove Locations

In a Works Process component, you have the option of removing one or all saved locations.

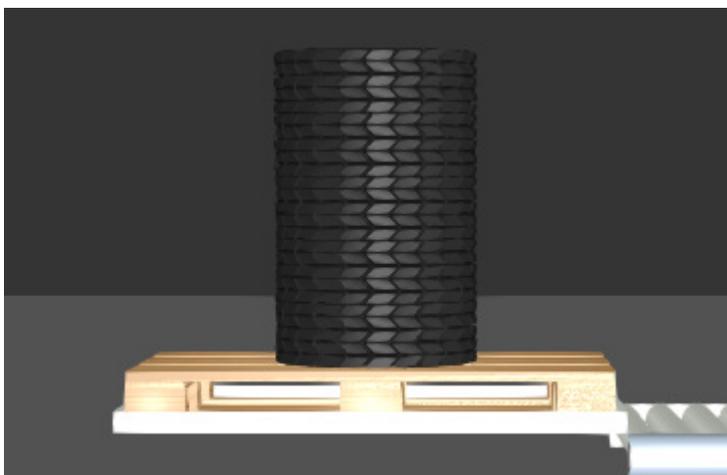
1. In the Component Properties panel, set Selection to **Euro Pallet**, and then click **RemoveSelected**.



2. Set EndRange to **9999**, and then click **ReplaceTask**. Now, all tires in the CreatePattern task will be created during a simulation.



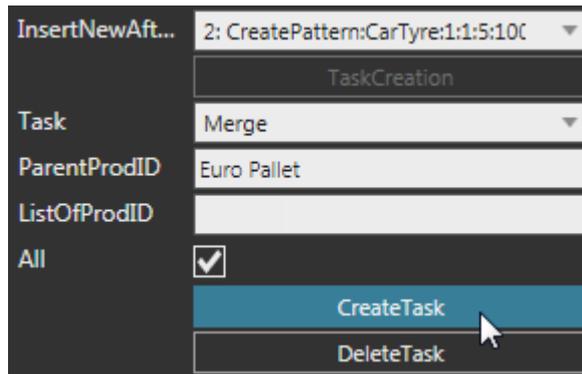
3. Run the simulation, verify a stack of tires is on top of a pallet, and then reset the simulation.



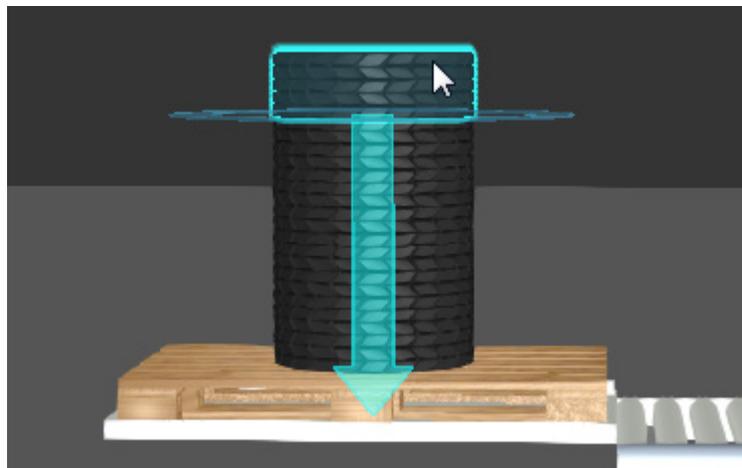
# Task - Merge

A Merge task allows you to attach components contained by a Works Process component to one another. This is helpful if you need groups of components to move together, for example a pallet with a stack of parts. The attachment of components is done by using ProdID values.

1. In the Component Properties panel, Default tab, set Task to **Merge**, ParentProdID to **Euro Pallet**, and then click **CreateTask**. This will attach everything, for example the tires, contained in the Works Process component to the pallet.



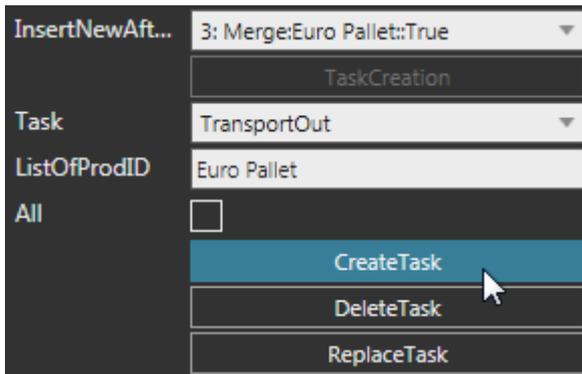
2. Run the simulation, select one of the **tires** in the stack, and then stop the simulation. With the PnP command enabled, verify the selected tire is attached to the pallet, and then reset the simulation.



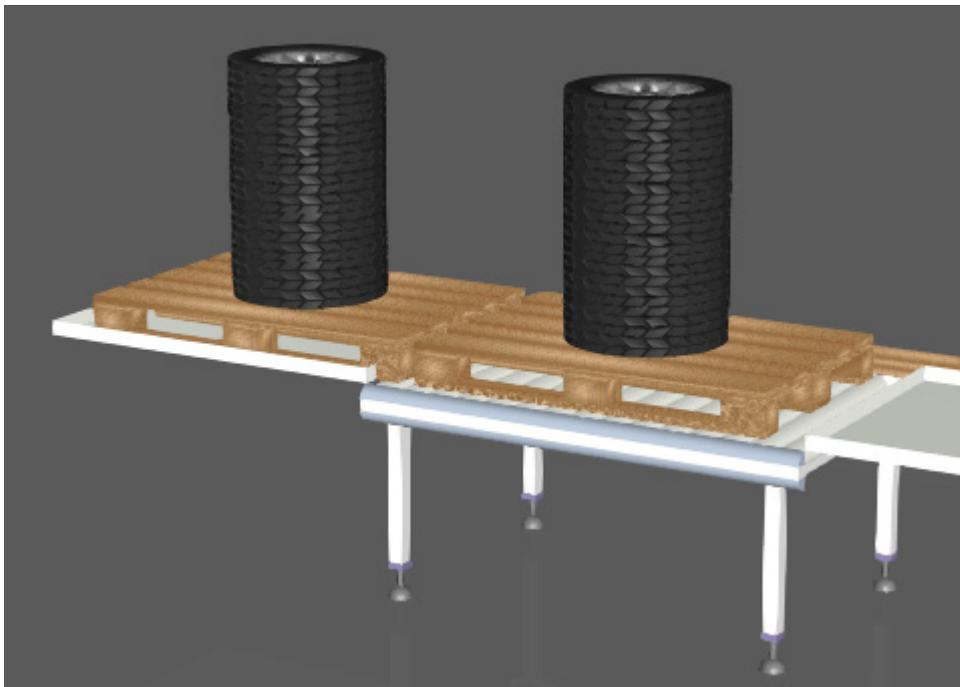
# Task - Transport Out

A Transport Out task allows you to move components out of a Works Process component via its path. You can transfer out all contained components or filter them by ProdID.

1. In the 3D world, select the **WorksProcess**.
2. In the Component Properties panel, Default tab, set Task to **TransportOut**, ListOfProdID to **Euro Pallet**, clear the **All** check box, and then click **CreateTask**. Since the preceding task merges tires with the pallet, transporting out the pallet will also move its child components.



3. Run the simulation, verify the pallet and tires move onto a conveyor, and then reset the simulation.

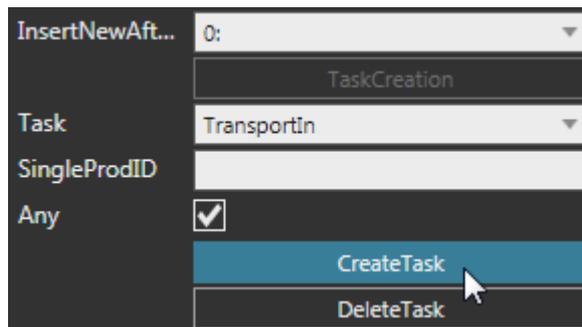


**NOTE!** A new stack of tires and pallet are created in the Works Process component because its process will reset and run again after completing the last task. The number of times a process is executed is controlled by the Task::RunTaskTimes property.

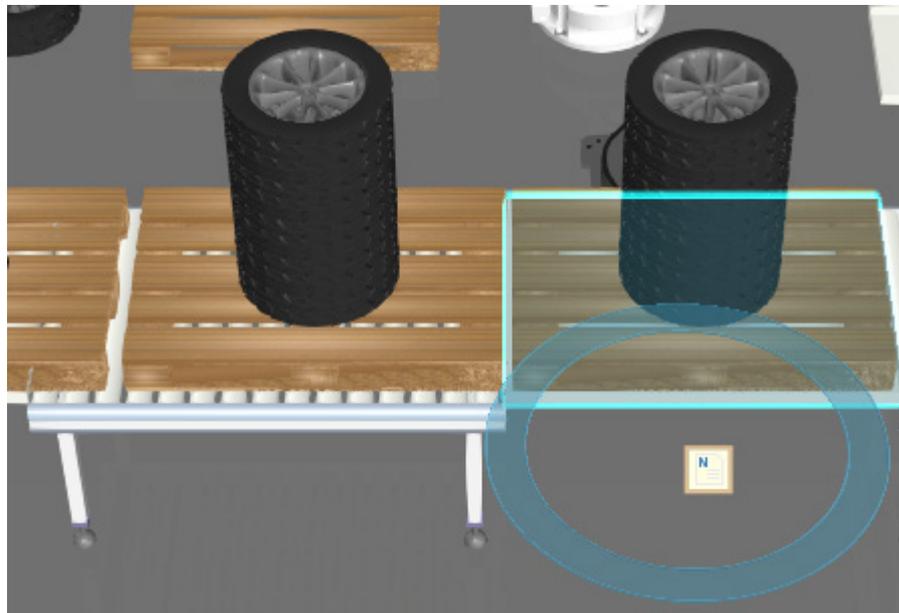
## Task - Transport In

A Transport In task allows you to move components into a Works Process component via its path and stop them. You can transfer in any component or filter them by ProdID. Generally, you transport in components to process them. Otherwise, a Works Process component will act like a conveyor, transfer in and out components via its path without processing them.

1. In the Cell Graph panel, under the Works category, click **WorksProcess #2** to select that component in the 3D world.
2. In the Component Properties panel, Default tab, set Task to **TransportIn**, and then click **CreateTask**.



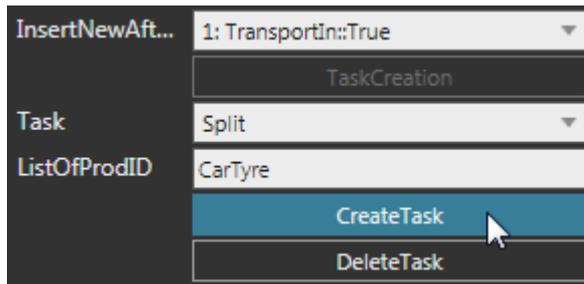
3. Run the simulation, verify a pallet of tires stops inside WorksProcess #2, and then reset the simulation.



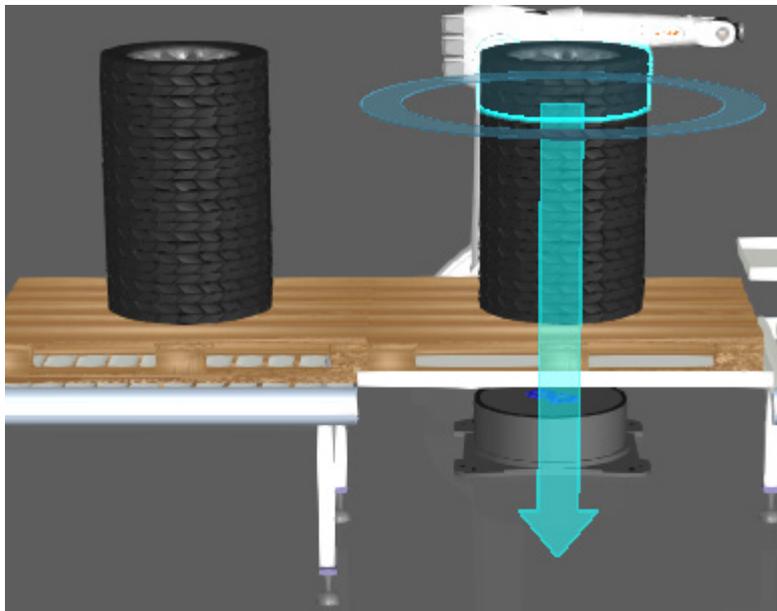
# Task - Split

A Split task allows you to detach components contained by a Works Process component from one another. This is helpful if you need to ungroup stacks of components in order to pick and place them separately. The detachment of components is done by using ProdID values.

1. In the Component Properties panel, Default tab, set Task to **Split**, ListOfProdID to **CarTyre**, and then click **CreateTask**. This will detach the tires from the pallet, thereby the tires will be attached to Works Process #2.



2. Run the simulation, wait until a pallet stops at WorksProcess #2, and then stop the simulation.
3. Select one of the **tires** in the stack at WorksProcess #2, and with the PnP command enabled, verify the selected tire is not attached to the pallet, and then reset the simulation.

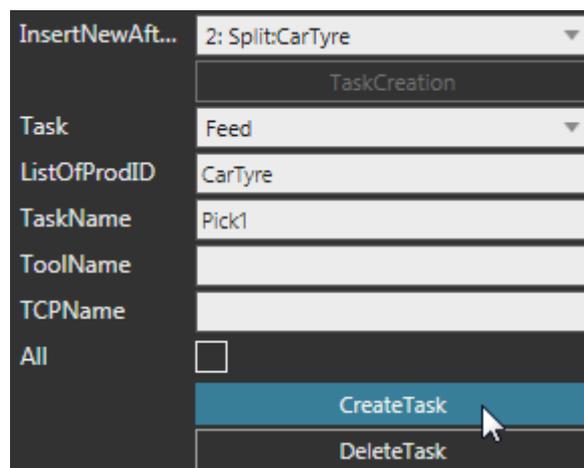


## Task - Feed

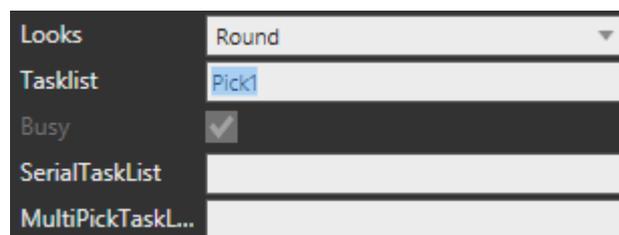
A Feed task allows you to signal resources that a list of components contained by a Works Process component are available and can be picked up and placed at other Works Process components. A TaskName is used to uniquely identify and assign a Feed task to a resource. ToolName and TCPName are used to identify a tool component and tool frame that a resource should use to perform the task. You can feed all contained components or filter them by ProdID.

The completion of a Feed task is when all of its fed components have been picked up by a resource. In order for a resource to pick up a component, there must be a need for the component at another Works Process component.

1. In the 3D world, select **WorksProcess #2**.
2. In the Component Properties panel, Default tab, set Task to **Feed**, TaskName to **Pick1**, clear the **All** check box, and then click **CreateTask**. This will feed the tires not the pallet.
3. In the 3D world, plug the **robot** to the **Works Robot Controller**. The controller is used to assign tasks to a connected robot used as a resource. In addition, the controller automates the program of a connected robot to complete assigned tasks.



4. Select the **WorksRobotController**, and then in the Component Properties panel, Default tab, set Tasklist to **Pick1**.



5. Run the simulation, verify no more pallets are transported into WorksProcess #2 because its Feed task is not completed, and then reset the simulation.

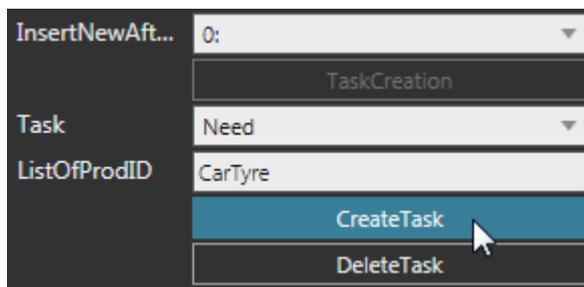
# Task - Need

A Need task allows you to request a list of components be placed in a Works Process component. The request is done by using ProdID values.

The setup of a Need task is rather simple, but requires other Works Process components to feed the needed products. That is, a resource must know from where the needed components can come from in order to deliver them.

The completion of a Need task is when all of the listed components by ProdID have been placed by a resource.

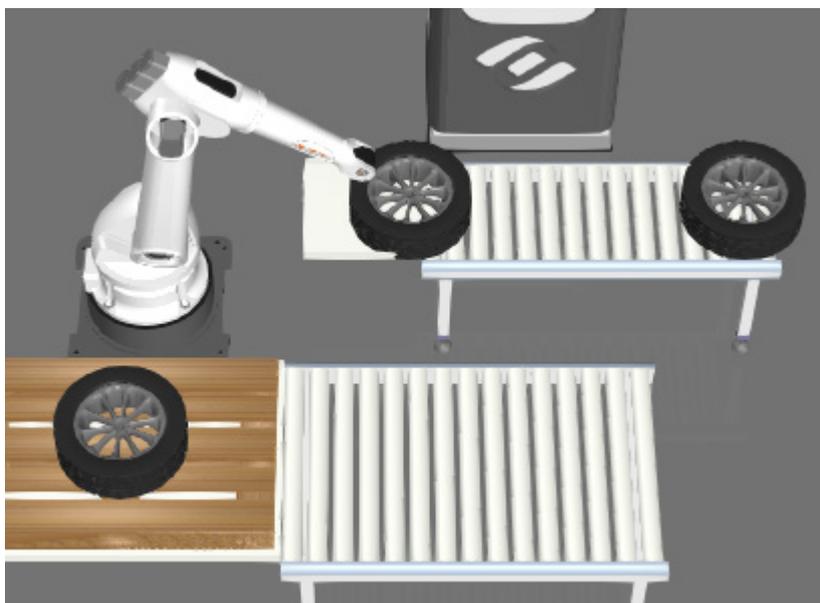
1. In the Cell Graph panel, under the Works category, click **WorksProcess #3** to select that component in the 3D world.
2. In the Component Properties panel, Default tab, set Task to **Need**, ListOfProdID to **CarTyre**, and then click **CreateTask**.



3. Set Task to **TransportOut**, and then click **CreateTask**. This means a tire will be transferred out as soon as it is received at WorksProcess #3.

**NOTE!** For a TransportOut task, the All property overrides the ListOfProdID property.

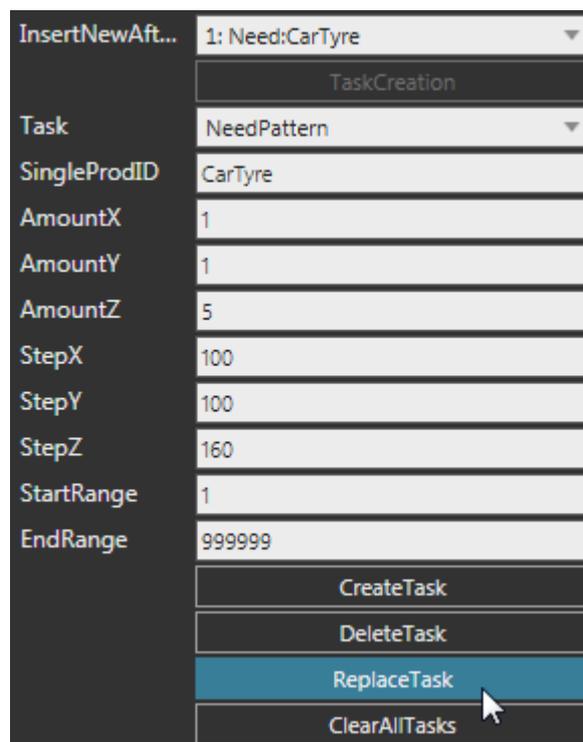
4. Run the simulation, verify the robot picks and places tires from WorksProcess #2 to WorksProcess #3, and then reset the simulation.



## Task - Need Pattern

A Need Pattern task allows you to request a pattern of components be placed in a Works Process component. Similar to a Need task, the needed component is identified by its ProdID and must come from other Works Process components.

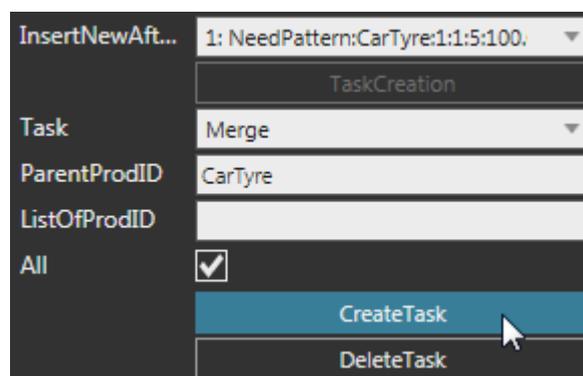
1. In the Component Properties panel, Default tab, set InsertNewAfterLine to the **first task** in the process, which requests the need of a tire.
2. Set Task to **NeedPattern**, SingleProdID to **CarTyre**, AmountX and Y to **1**, AmountZ to **5**, StepZ to **160**, and then click **ReplaceTask**. Now, there is a need for a stack of 5 tires.



The screenshot shows the Component Properties panel with the following settings:

InsertNewAft...	1: Need:CarTyre
Task	NeedPattern
SingleProdID	CarTyre
AmountX	1
AmountY	1
AmountZ	5
StepX	100
StepY	100
StepZ	160
StartRange	1
EndRange	999999
	CreateTask
	DeleteTask
	ReplaceTask
	ClearAllTasks

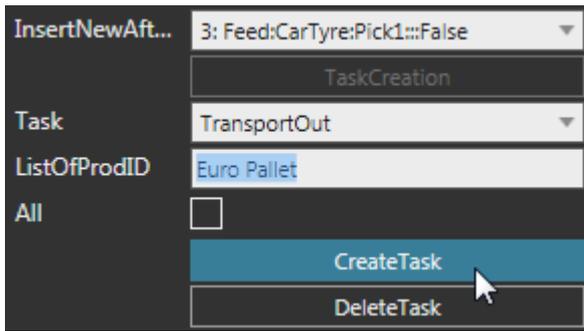
3. In the Component Properties panel, Default tab, set Task to **Merge**, ParentProdID to **CarTyre**, clear **ListOfProdID**, and then click **CreateTask**. This will attach the stack of tires to one another, thereby allowing the stack to move as a group.



The screenshot shows the Component Properties panel with the following settings:

InsertNewAft...	1: NeedPattern:CarTyre:1:1:5:100
Task	Merge
ParentProdID	CarTyre
ListOfProdID	
All	<input checked="" type="checkbox"/>
	CreateTask
	DeleteTask

- In the 3D world, select **WorksProcess #2**, and then add a **TransportOut** task for moving out empty pallets.



- Run the simulation, verify the robot picks and places a stack of tires, and then reset the simulation.



## Review

In this tutorial you learned how to simulate simple processes using components in the Works library. For example, you created processes for depalletizing products using a robot. You understand the basics of creating and editing tasks in processes. This will help you in other Works library tutorials covering more complex processes and tasks. As a guide, we recommend you now learn more about ProdID filtering and serial task lists for robots.