Business Workflows

11.2

February 2026

DOCUMENT ACCESS

Public

DISCLAIMER

The contents of this document are under copyright of Critical Manufacturing S.A. it is released on condition that it shall not be copied in whole, in part or otherwise reproduced (whether by photographic, or any other method) and the contents therefore shall not be divulged to any person other than that of the addressee (save to other authorized offices of his organization having need to know such contents, for the purpose for which disclosure is made) without prior written consent of submitting company.

Business Workflows

Business Workflows is a tool within Critical Manufacturing MES that allows users to design DEEs using a low-code visual interface with drag-and-drop tasks. It is ideal for building workflows to address simple scenarios and is managed as a system-versioned entity.

These applications can be used for standalone functions or event handling. In this tutorial, we'll focus on the event handling use case.

This tool is designed for users who are not IT experts, but who have a basic understanding of workflows, which you will have at the end of this tutorial.

Standalone or Business Event

Overview

In this tutorial, our goal is to walk through three progressively complex scenarios, each reflecting possible use cases in a client factory environment:

- Shipping a Material
- Preventing a Material from starting production
- Placing a retrieved Material to start production

Throughout these scenarios, we'll explore the following key concepts:

- Business Event Tasks
- Material, Resource and Notification MES tasks
- Reference Syntax
- Addition of inputs and setting override

Syntax

Before diving in, it is important to understand the syntax used to reference data that will be used as inputs in your workflow.

Each task available in the workflow has a unique name, along with its defined inputs and outputs. To reference this data, there is a structure to be followed.

If you want to access the input of a given task, this is the syntax used: `{{$task_name.input_or_output_name}}`.

- Example: `{{$MaterialTrackOutPost_1.Material}}`

If you want to access properties within the input of a give task, this is the syntax used: `{{$task_name.input_or_output_name.property_1.property_2.[...].property_n}}`.

- Example: `{{$MaterialTrackOutPost_1.Material.LastProcessedResource.Type}}`

To know more about this, check out Business Workflow Task.

Scenario: Ship Material

Consider a production context with multiple production **Flows**, where the final **Step** in each **Flow** is a pass-through **Step** with the Allow Shipping flag set to true.

The requirement is: when a **Material** reaches these **Steps**, it should be automatically shipped to a new **Facility** - the Distribution Center.

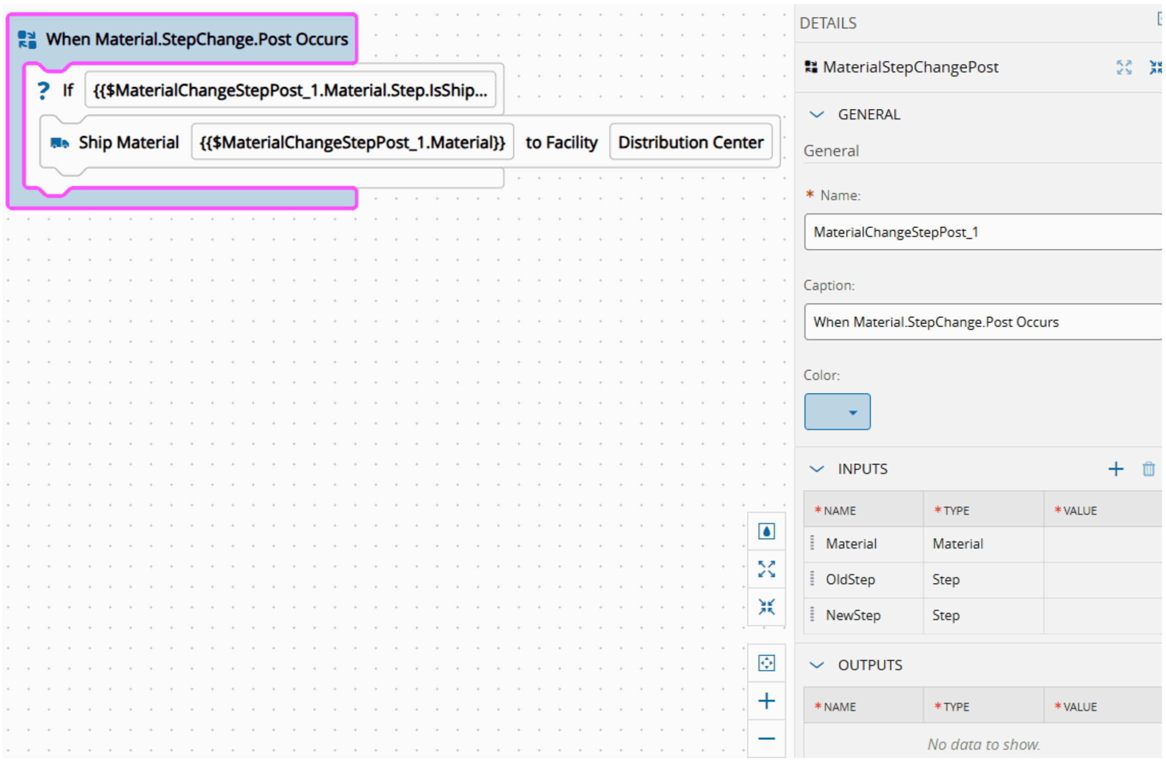
Configuration

To configure this scenario in the system, we start by identifying the trigger. What initiates the workflow?

The answer is: when the **Material** changes **Step**. This leads us to select the Business Workflow event **Material.ChangeStep.Post.**

Next, the system must evaluate the current **Step** the **Material** has reached. This highlights the need for a **Logic Task If**, where the condition checks if the Is Shipping Allowed property of the **Step** is set to true. If so, the **Material** should be shipped to the Distribution Center, which requires the use of the **MES Material Task Ship**.

Follow the tabs below, from left to right, to navigate through the resulting configuration:



When Material.StepChange.Post Occurs

If {{$MaterialChangeStepPost_1.Material.Step.IsShip...

Ship Material {{$MaterialChangeStepPost_1.Material}} to Facility Distribution Center

DETAILS

MaterialStepChangePost

GENERAL

General

* Name: MaterialChangeStepPost_1

Caption: When Material.StepChange.Post Occurs

Color:

INPUTS

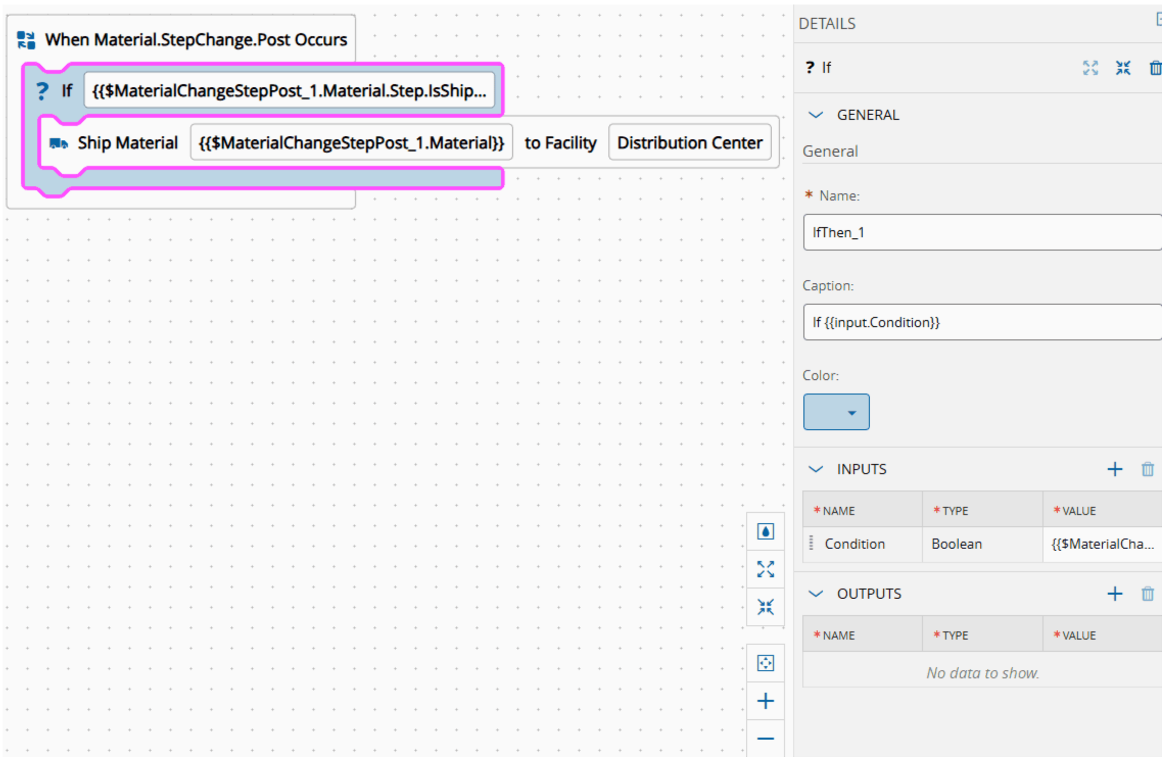
*NAME	*TYPE	*VALUE
Material	Material	
OldStep	Step	
NewStep	Step	

OUTPUTS

*NAME	*TYPE	*VALUE

No data to show.

Logic Task If



When Material.StepChange.Post Occurs

If {{$MaterialChangeStepPost_1.Material.Step.IsShip...

Ship Material {{$MaterialChangeStepPost_1.Material}} to Facility Distribution Center

DETAILS

? If

GENERAL

General

* Name: IfThen_1

Caption: If {{input.Condition}}

Color:

INPUTS

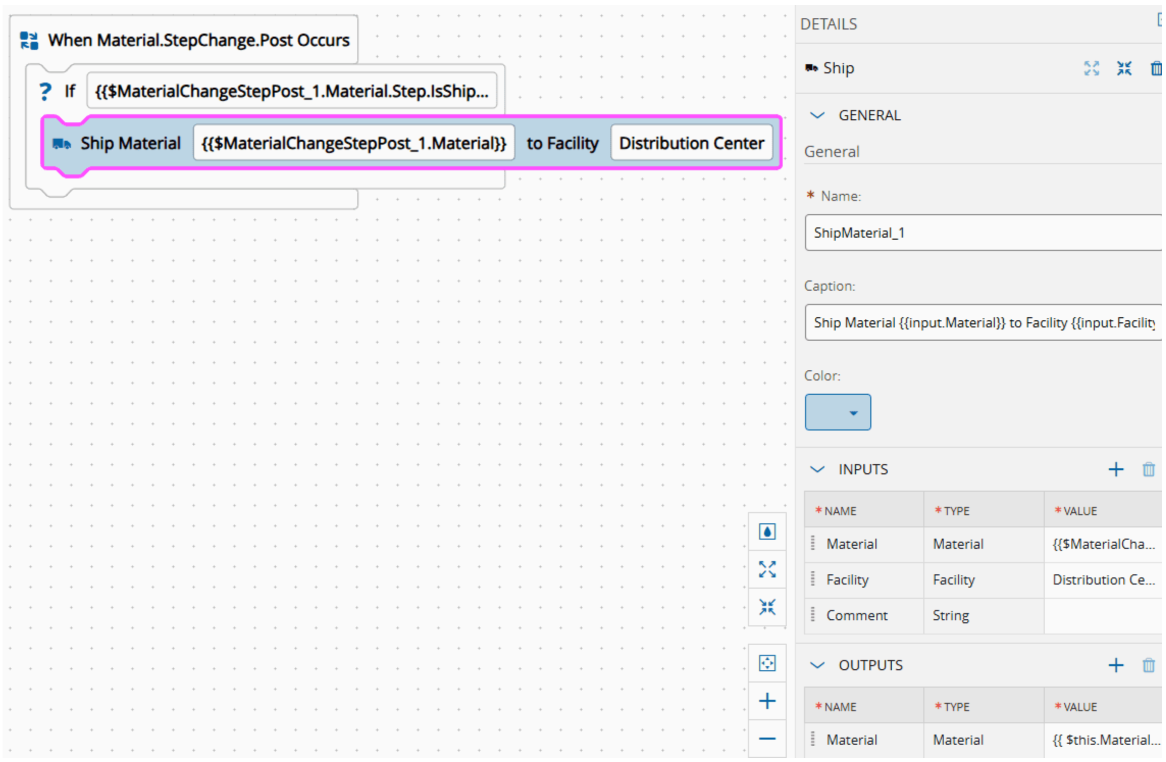
*NAME	*TYPE	*VALUE
Condition	Boolean	{{$MaterialCha...

OUTPUTS

*NAME	*TYPE	*VALUE

No data to show.

MES Material Task Ship



When Material.StepChange.Post Occurs

If {{$MaterialChangeStepPost_1.Material.Step.IsShip...

Ship Material {{$MaterialChangeStepPost_1.Material}} to Facility Distribution Center

DETAILS

Ship

GENERAL

General

* Name: ShipMaterial_1

Caption: Ship Material {{input.Material}} to Facility {{input.Facility

Color:

INPUTS

*NAME	*TYPE	*VALUE
Material	Material	{{$MaterialCha...
Facility	Facility	Distribution Ce...
Comment	String	

OUTPUTS

*NAME	*TYPE	*VALUE
Material	Material	{{ $this.Material...

Here's a step-by-step breakdown of the configuration process:

1. Start by creating a new Business Workflow. Select the Business Event best suited for this scenario:
   - **Material.ChangeStep.Post**
2. Drag and drop the **Logic Task If**, and set the following condition:
   - `{{$MaterialChangeStepPost_1.Material.Step.IsShippingAllowed==true}}`

3. Drag and drop the **MES Material Task Ship**, and configure its inputs:
   - Material: `{{$MaterialChangeStepPost_1.Material}}`
   - Facility: Distribution Center
4. Save and set this version as effective.

See more information about these tasks here

You can find the exportable file of this scenario here: XML File.

Testing

Preconditions:

- Ensure there is a **Flow** with a **Step** where shipping is not allowed — this is where the **Material** should initially be.
- Define another **Step** with the Is Shipping Allowed property set to true, either within the same **Flow** or in a different one.
- Confirm that **Facility** relationships are configured to allow shipping between the current location and the Distribution Center.

Test scenario:

- Perform a Move Next on a **Material**, where the next **Step** has shipping allowed.
- Alternatively, perform a Change **Flow** and **Step** to one where shipping is allowed.
- In both cases, the **Material** should transition to `In Transit`, indicating that it was successfully shipped.

Scenario: Prevent Material from Starting Production

Consider a manufacturing scenario where a minimum of 10 units of the primary quantity of the Material is required for production to begin.

The requirement is: if a **Material** is dispatched and this minimum is not met, it should automatically be placed on hold with the "Quantity Below Minimum" **Reason**, and the **Resource** should transition to the `Unscheduled Down` state.

Configuration

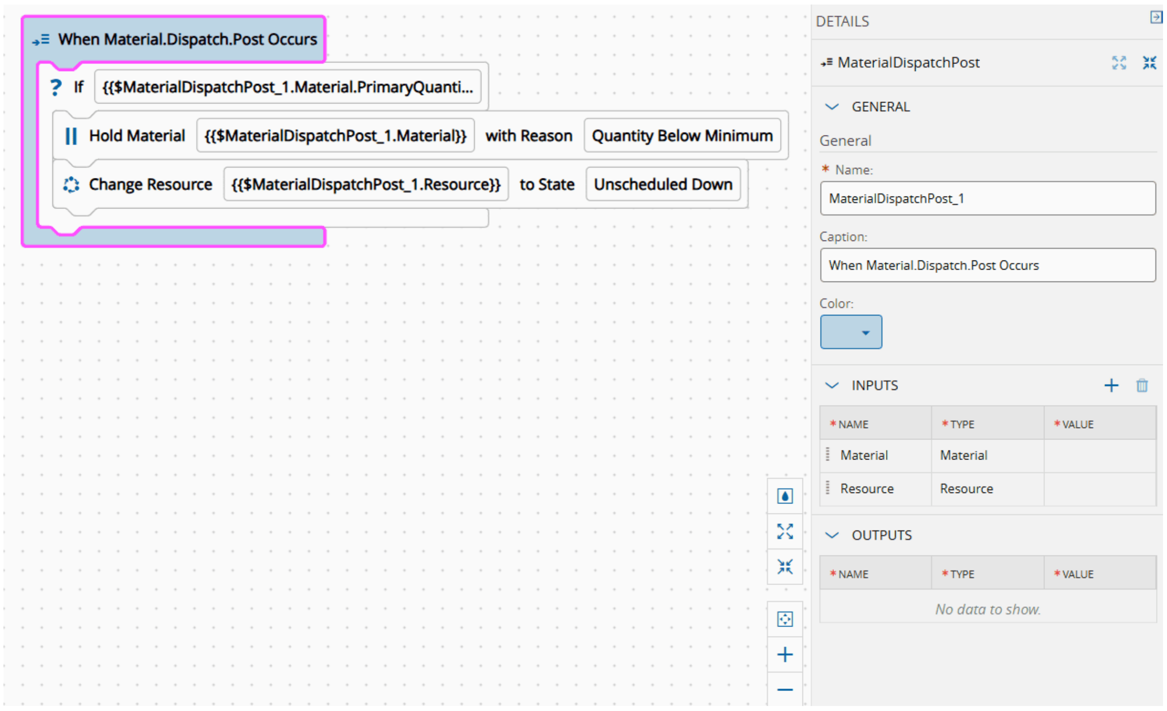
The trigger for this scenario is the dispatching of a **Material**, which means the appropriate Business Event for this workflow is **Material.Dispatch.Post**.

Since there is a condition based on the primary quantity of the **Material**, we need to use a **Logic Task If** to evaluate whether the quantity is less than 10. If the condition is met, two actions should occur:

- The **Material** should be placed on hold, using the **MES Material Task Hold** task.
- The **Resource** where the **Material** was dispatched to should be set to the `Unscheduled Down` state, using the **MES Resource Task Change State** task.

Follow the tabs below, from left to right, to navigate through the resulting configuration:

Business Event Task



When Material.Dispatch.Post Occurs

If {{$MaterialDispatchPost_1.Material.PrimaryQuanti...

Hold Material {{$MaterialDispatchPost_1.Material}} with Reason Quantity Below Minimum

Change Resource {{$MaterialDispatchPost_1.Resource}} to State Unscheduled Down

DETAILS

MaterialDispatchPost

GENERAL

General

* Name: MaterialDispatchPost_1

Caption: When Material.Dispatch.Post Occurs

Color:

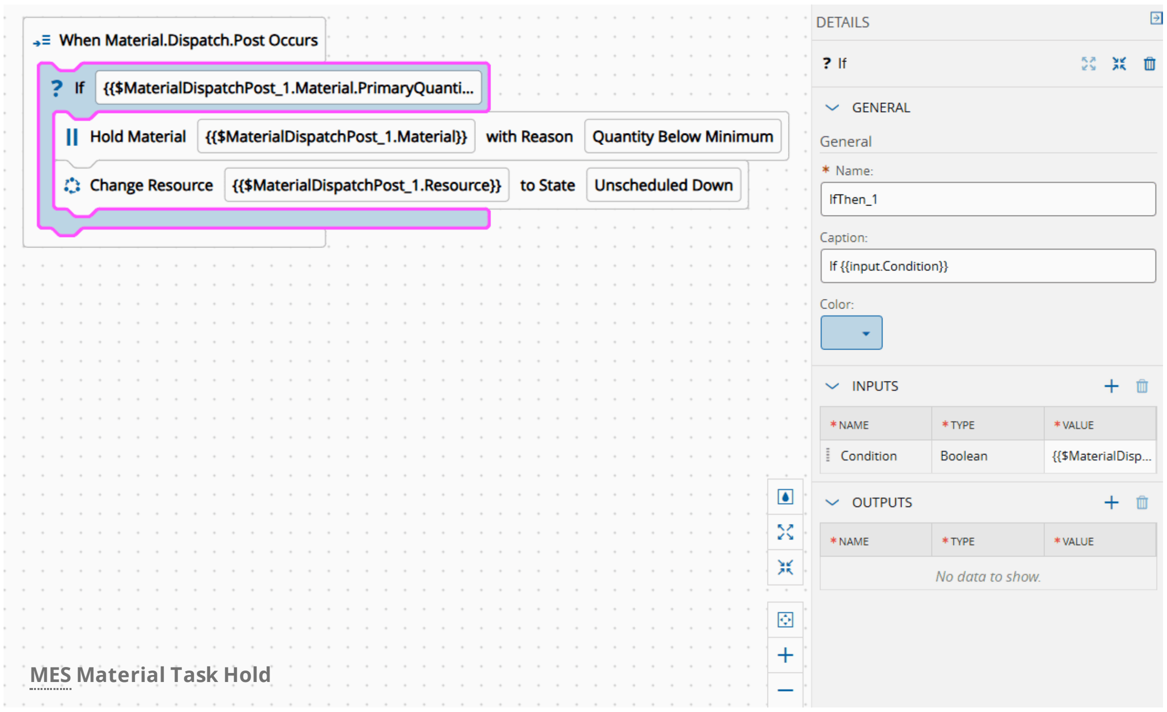
INPUTS

* NAME	* TYPE	* VALUE
Material	Material	
Resource	Resource	

OUTPUTS

* NAME	* TYPE	* VALUE
No data to show.		

Logic Task If



When Material.Dispatch.Post Occurs

? If {{$MaterialDispatchPost_1.Material.PrimaryQuanti...

Hold Material {{$MaterialDispatchPost_1.Material}} with Reason Quantity Below Minimum

Change Resource {{$MaterialDispatchPost_1.Resource}} to State Unscheduled Down

MES Material Task Hold

DETAILS

? If

GENERAL

General

* Name: IfThen_1

Caption: If {{input.Condition}}

Color:

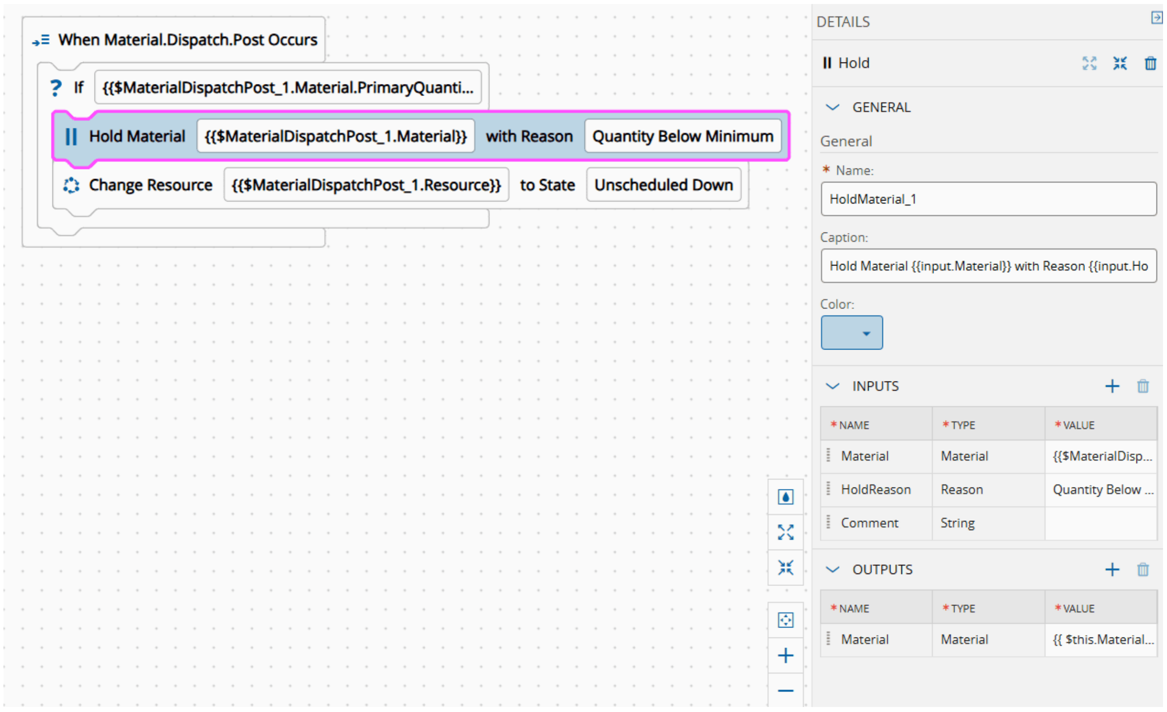
INPUTS

* NAME	* TYPE	* VALUE
Condition	Boolean	{{$MaterialDisp...

OUTPUTS

* NAME	* TYPE	* VALUE

No data to show.



When Material.Dispatch.Post Occurs

? If {{$MaterialDispatchPost_1.Material.PrimaryQuanti...

Hold Material {{$MaterialDispatchPost_1.Material}} with Reason Quantity Below Minimum

Change Resource {{$MaterialDispatchPost_1.Resource}} to State Unscheduled Down

DETAILS

Hold

GENERAL

General

* Name: HoldMaterial_1

Caption: Hold Material {{input.Material}} with Reason {{input.Ho

Color:

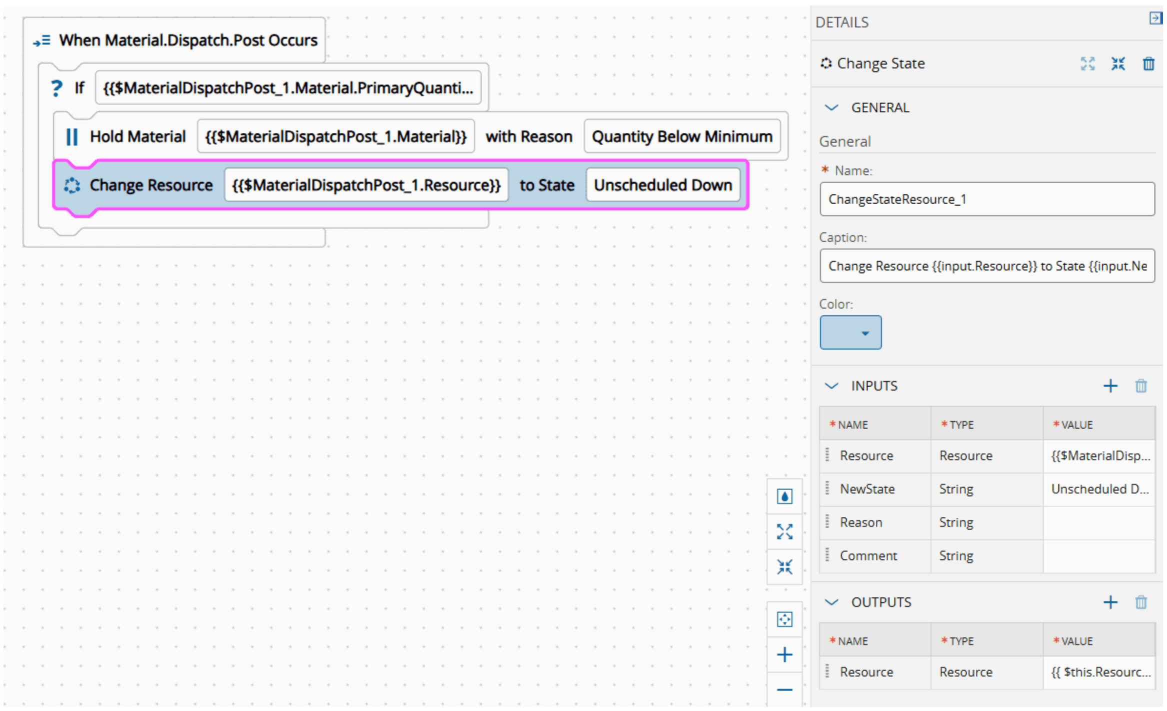
INPUTS

* NAME	* TYPE	* VALUE
Material	Material	{{$MaterialDisp...
HoldReason	Reason	Quantity Below ...
Comment	String	

OUTPUTS

* NAME	* TYPE	* VALUE
Material	Material	{{ $this.Material...

MES Resource Task Change State



When Material.Dispatch.Post Occurs

If {{$MaterialDispatchPost_1.Material.PrimaryQuanti...

Hold Material {{$MaterialDispatchPost_1.Material}} with Reason Quantity Below Minimum

Change Resource {{$MaterialDispatchPost_1.Resource}} to State Unscheduled Down

DETAILS

Change State

GENERAL

General

* Name: ChangeStateResource_1

Caption: Change Resource {{input.Resource}} to State {{input.Ne

Color:

INPUTS

* NAME	* TYPE	* VALUE
Resource	Resource	{{$MaterialDisp...
NewState	String	Unscheduled D...
Reason	String	
Comment	String	

OUTPUTS

* NAME	* TYPE	* VALUE
Resource	Resource	{{ $this.Resourc...

Here's a step-by-step breakdown of the configuration process:

1. Start by creating a new Business Workflow. Select the Business Event best suited for this scenario: **Material.Dispatch.Post**
2. Drag and drop the **Logic Task If**, and set the following condition: `{{$MaterialDispatchPost_1.Material.PrimaryQuantity<10}}`
3. Drag and drop the **MES Material Task Hold**, and configure its inputs:
   - Material: `{{$MaterialDispatchPost_1.Material}}`
   - Reason: Quantity Below Minimum
4. Drag and drop the **MES Resource Task Change State**, and configure its inputs:

- Resource: `{{$MaterialDispatchPost_1.Resource}}`
- State: Unscheduled Down

5. Save and set this version as effective.

See more information about these tasks here

You can find the exportable file of this scenario here: XML File.

Testing

Preconditions:

- The **Resource** to be used must be in a state other than `Unscheduled Down`.
- The Hold **Reason** must be created and associated with the **Step** being used.
- The primary quantity of the **Material** must be less than 10.

Test scenario:

- Perform a **Material** Dispatch.
- After dispatching, the **Material** should be placed on hold with the appropriate Reason, and the Resource should transition to the `Unscheduled Down` state.

Scenario: Place Retrieved Material to Start Production

Consider the scenario where the requirements are: if a **Material** is retrieved from the Warehouse **Step**, it should automatically be moved to the **Step** where production starts, by updating its **Flow** and **Step**.

Additionally, a **Notification** should be sent indicating that the **Material** is ready for production. This notification must have the information about the **Product**, **Step**, and Facility associated with the **Material**.

Configuration

The trigger for this scenario is when a **Material** is retrieved, which means the appropriate Business Event is **Material.Retrieve.Post**.

A condition must then be evaluated to ascertain whether the **Step** from which the **Material** is retrieved is the Warehouse **Step**. This requires the use of the **Logic Task If** task. If the condition is met, two actions should follow:

- The **Flow** and **Step** of the **Material** should be updated, which calls for the use of the **MES Material Task Change Flow and Step** task.
- A **Notification** should be sent, using the **MES Notification Task Create Notification** task. In this case, we'll use a general notification that dynamically displays the **Material** name, along with its **Product**, **Step**, and **Facility**.

To achieve this dynamic content, we'll introduce a new concept: adding inputs to override task settings — such as the message content in the Create Notification task.

Adding Inputs and Outputs

To add new inputs or outputs, use the + icon located on the left pane of the designer, on the input or output section.

After clicking it:

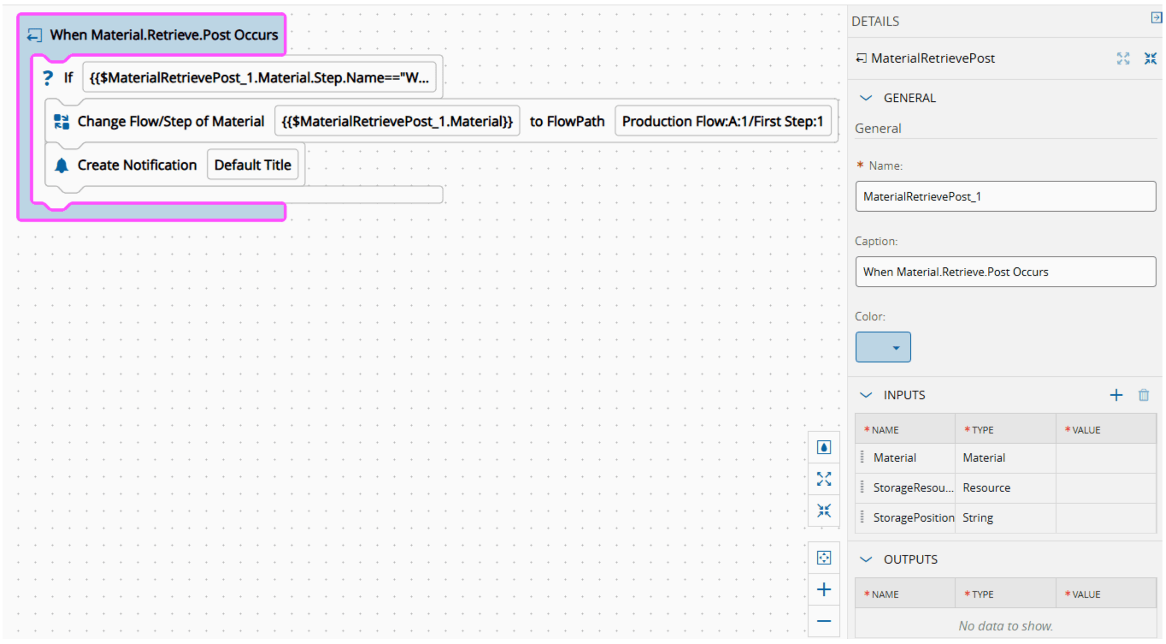
- Provide a name that starts with a letter and does not contain whitespaces or special characters.
- Then, select the appropriate Type from the list of available options.
- Lastly, define the Collection Type.

> ⚠ **Warning**
>
> The data the Collection Type "Array" is currently not functional - this is due to the shared structure and logic between Connect IoT features and the Business Workflow feature.

To know more about this, check out Business Workflow Task.

Going back to the scenario, follow the tabs from left to right to see the resulting configuration for the first three tasks:

Business Event Task



When Material.Retrieve.Post Occurs
If {{$MaterialRetrievePost_1.Material.Step.Name=="W...
Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1
Create Notification Default Title

DETAILS

MaterialRetrievePost

GENERAL

General

* Name: MaterialRetrievePost_1

Caption: When Material.Retrieve.Post Occurs

Color:

INPUTS

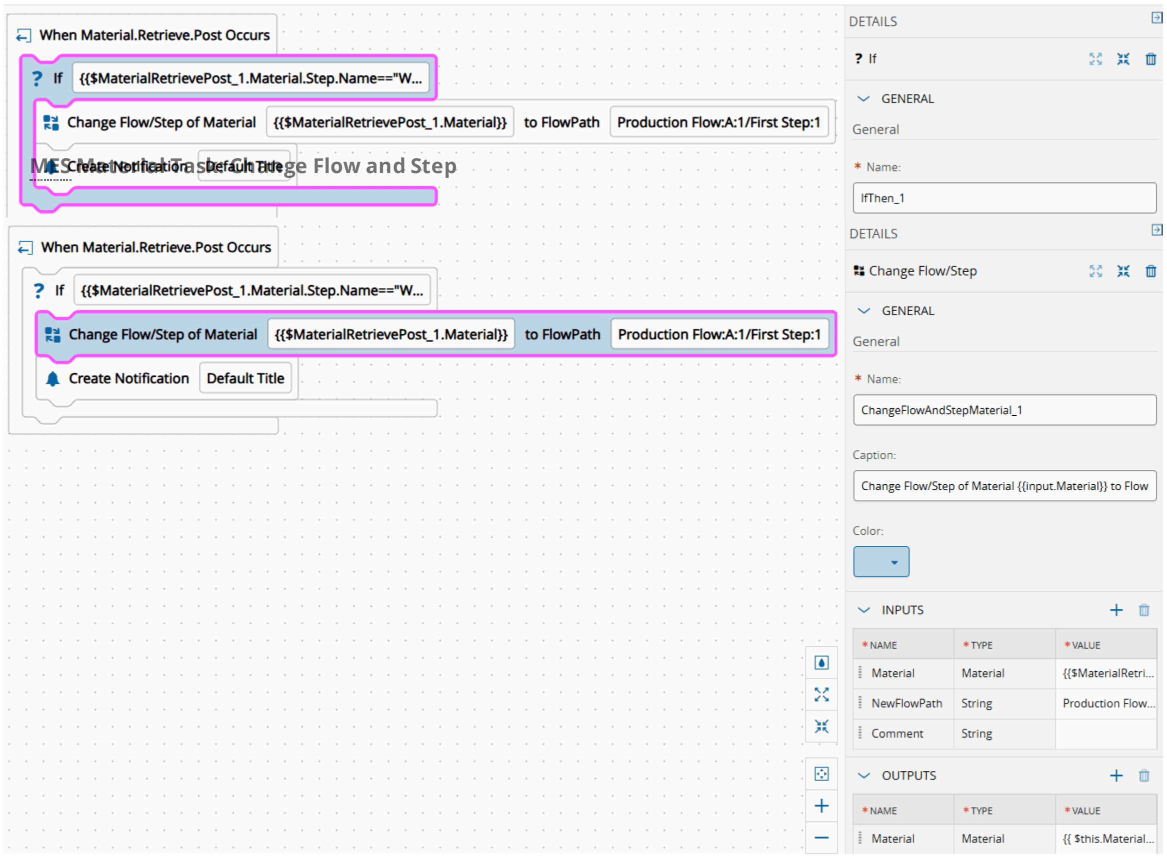
*NAME	*TYPE	*VALUE
Material	Material	
StorageResou...	Resource	
StoragePosition	String	

OUTPUTS

*NAME	*TYPE	*VALUE
No data to show.		

Logic Task If

MES Material Task Change Flow and Step



When Material.Retrieve.Post Occurs

? If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

Create Notification Default Title

DETAILS

? If

GENERAL

General

* Name: IfThen_1

When Material.Retrieve.Post Occurs

? If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

Create Notification Default Title

DETAILS

Change Flow/Step

GENERAL

General

* Name: ChangeFlowAndStepMaterial_1

Caption: Change Flow/Step of Material {{input.Material}} to Flow

Color:

INPUTS

* NAME	* TYPE	* VALUE
Material	Material	{{$MaterialRetri...
NewFlowPath	String	Production Flow...
Comment	String	

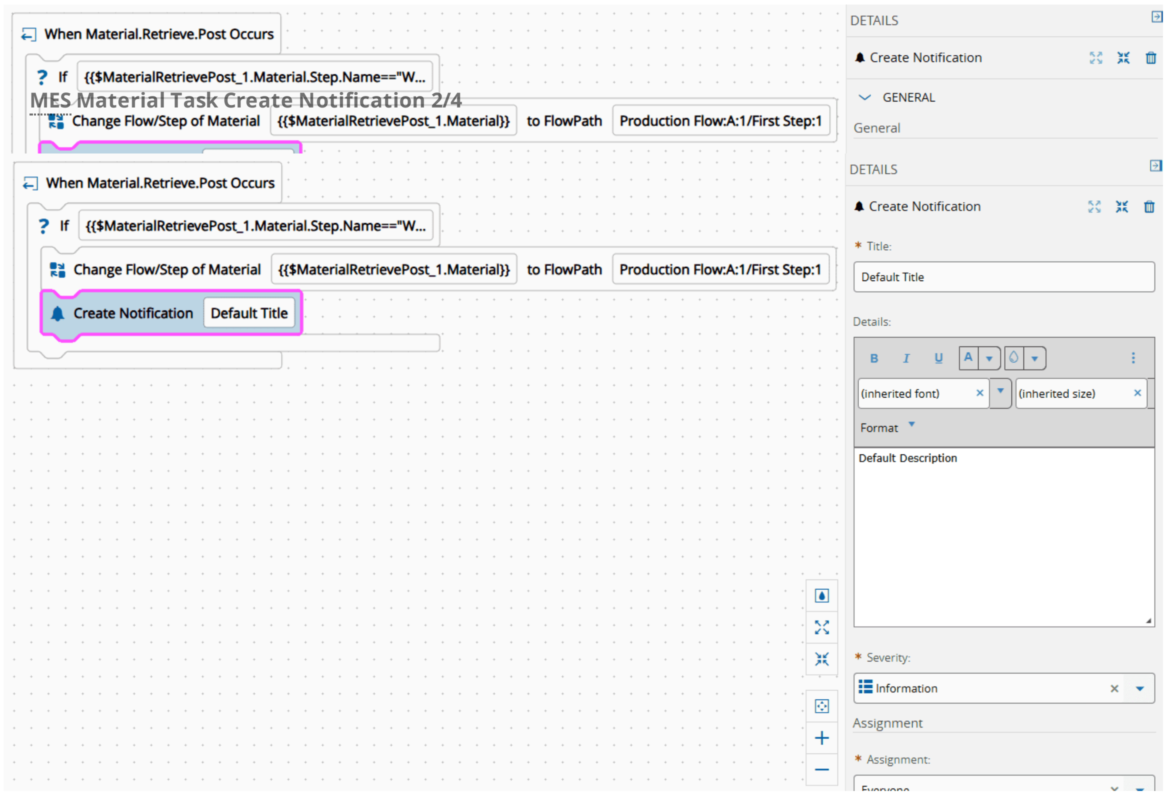
OUTPUTS

* NAME	* TYPE	* VALUE
Material	Material	{{ $this.Material...

To make it easier to read, you can see continue following the tabs from left to right to see the configuration of the **MES Material Task Create Notification** task:

MES Material Task Create Notification 1/4

MES Material Task Create Notification 2/4



When Material.Retrieve.Post Occurs

? If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

DETAILS

Create Notification

GENERAL

General

When Material.Retrieve.Post Occurs

? If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

Create Notification Default Title

DETAILS

Create Notification

* Title: Default Title

Details:

(inherited font) (inherited size)

Format

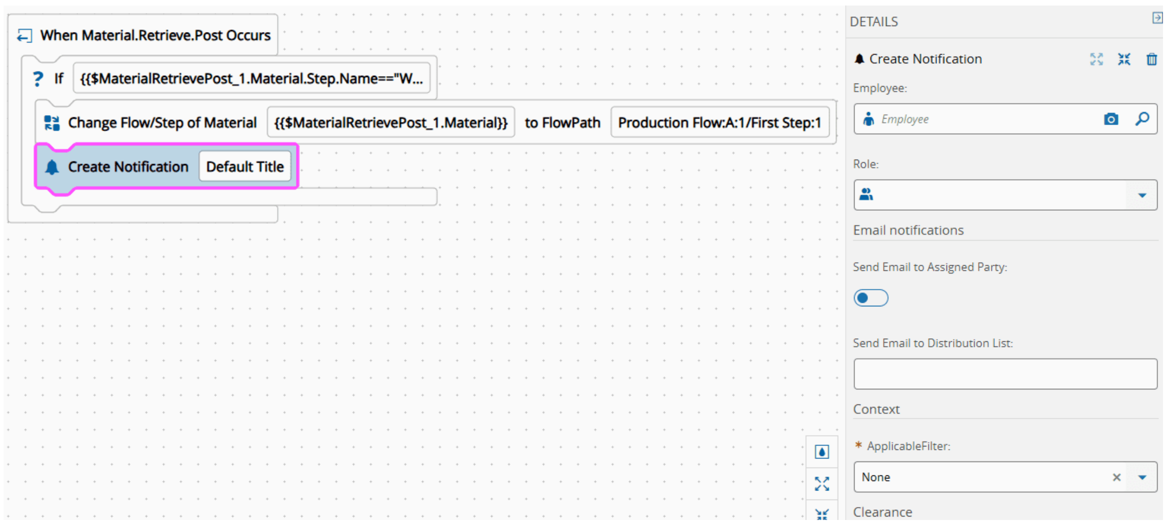
Default Description

* Severity: Information

Assignment

* Assignment: Everyone

MES Material Task Create Notification 3/4



When Material.Retrieve.Post Occurs

? If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

Create Notification Default Title

DETAILS

Create Notification

Employee: Employee

Role:

Email notifications

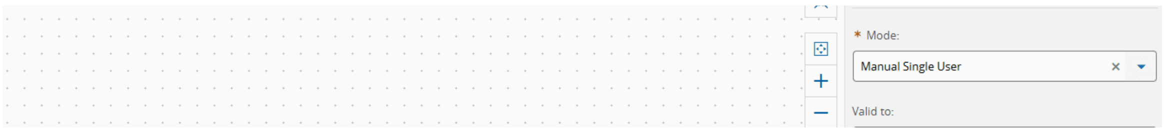
Send Email to Assigned Party:

Send Email to Distribution List:

Context

* ApplicableFilter: None

Clearance

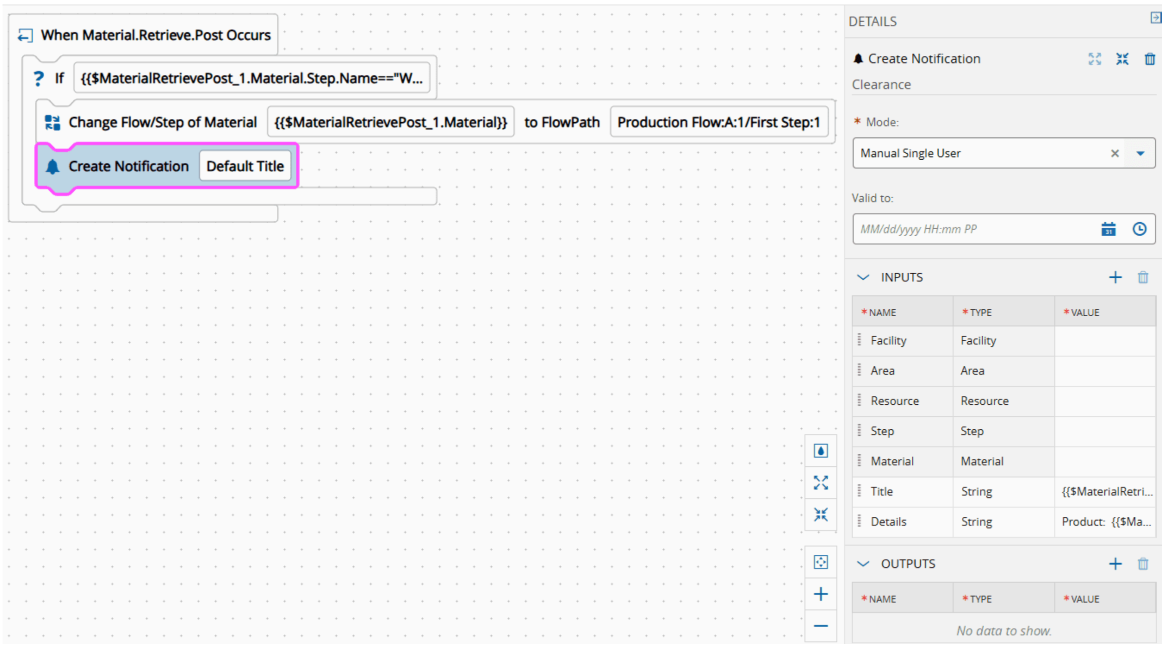


* Mode:

Manual Single User

Valid to:

MES Material Task Create Notification 4/4



When Material.Retrieve.Post Occurs

If {{$MaterialRetrievePost_1.Material.Step.Name=="W...

Change Flow/Step of Material {{$MaterialRetrievePost_1.Material}} to FlowPath Production Flow:A:1/First Step:1

Create Notification Default Title

DETAILS

Create Notification

Clearance

* Mode:

Manual Single User

Valid to:

MM/dd/yyyy HH:mm PP

INPUTS

*NAME	*TYPE	*VALUE
Facility	Facility	
Area	Area	
Resource	Resource	
Step	Step	
Material	Material	
Title	String	{{$MaterialRetri...
Details	String	Product: {{$Ma...

OUTPUTS

*NAME	*TYPE	*VALUE
No data to show.		

`{{$MaterialRetrievePost_1.Material.Step.Name=="Warehouse"}}`

3. Drag and drop the **MES Material Task Change Flow and Step** task, and configure its inputs:
   - Material: `{{$MaterialRetrievePost_1.Material}}`
   - Flow Path: Production Flow:A:1/First Step:1
4. Drag and drop the **MES Notification Task Create Notification** task, and configure its settings and inputs:
   - Settings:
     - Name: *[Empty]*
     - Description: Material Ready For Production
     - Type: General (value must exist on NotificationType Lookup Table)
     - Title: Default title (to be overriden)
     - Description: Default Description (to be overriden)
     - Severity: Information (value must exist on NotificationSeverity Lookup Table)
     - Assignment: Everyone
     - applicable filer: None
     - Mode: Manual Single User
   - Inputs:

- Add input
  - Name: Title
  - Type: String
  - Collection Type: None
  - Value: " `{{$MaterialRetrievePost_1.Material.Name}}` Material is ready for Production."
- Add input
  - Name: Details
  - Type: String
  - Collection Type: None
  - Value: "Product: `{{$MaterialRetrievePost_1.Material.Product.Name}}` `<br\>` Step Name: `{{$MaterialRetrievePost_1.Material.Step.Name}}` `<br\>` Facility Name: `{{$MaterialRetrievePost_1.Material.Facility.Name}}` "

5. Save and set this version as effective.

See more information about these tasks here

You can find the exportable file of this scenario here: XML File.

Testing

Preconditions:

- The Warehouse **Step** must have a required storage **Service**, and there must be a **Resource** available to provide it.
- The **Material** to be used should already be stored.
- The Generic Table NotificationSeverity must contain an entry for the configured **Notification** Type.

Test scenario:

- Perform **Material** Retrieve.
- After retrieval, the **Material** should move to the defined Flow Path, and a **Notification** should appear displaying the **Material** name, **Product**, **Step**, and **Facility**.

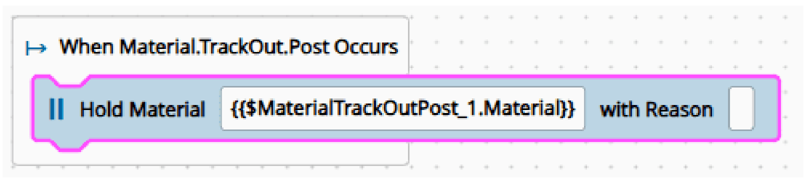
Final Notes

Caption

You can customize the caption of a task, which controls the text displayed on the block and its associated fields. While it is not mandatory and holds no functional impact, it can improve clarity and readability within the workflow.

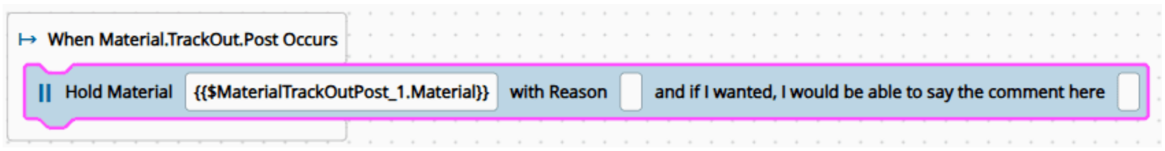
For example the default Caption for the **MES Material Task Hold** task is:

- Hold Material `{{input.Material}}` with `{{input.HoldReason}}` **Reason**.


When Material.TrackOut.Post Occurs
Hold Material
{{$MaterialTrackOutPost_1.Material}}
with Reason

However, you can modify this caption by adding or removing appearance inputs in the task block. Here's an example of a customized version:

- Hold Material `{{input.Material}}` with `{{input.HoldReason}}` **Reason** and if I wanted, I would be able to say the comment here `{{input.Comment}}`


When Material.TrackOut.Post Occurs
Hold Material
{{$MaterialTrackOutPost_1.Material}}
with Reason
and if I wanted, I would be able to say the comment here

For more information on caption, check out Business Workflow Task.

Input

Inputs are the data values required for a task (function) to execute. These values typically come from direct properties of entities referenced by "parent" tasks. No attributes or inherited Product Characteristics are available as inputs.

Inputs can be either referenced dynamically or hardcoded. For example, assuming the task has an input called "Resource", you can:

- Reference it dynamically: `{{$MaterialTrackOutPost_1.Resource}}`
- Hardcode the value: `Baker-01`

Both are valid, depending on the use case.

For more information on inputs and syntax, check out Business Workflow Task.

Note

To check the correct property names, go to: Administration, Entity Types, then select the entity you need and open the Properties section.

This is important because the name shown in the UI doesn't always match the actual property name.

For example, in the Material Details View, the last Resource where the material was processed appears as "Last Resource", but the property name is actually `LastProcessedResource`.

Output

Some tasks come with automatically defined outputs, such as `{{$this.Material}}`. These outputs represent the result of the task itself and may be required by other system processes or dependencies.

It is strongly advised not to remove these outputs, as doing so could impact the proper functioning of the workflow or system behavior as a whole.

This syntax is not used as an input in other tasks—it's specific to the task's own context.

For more information on outputs and syntax, check out Business Workflow Task.

Tasks

If you want to explore other tasks, checkout this Business Workflow Tasks.

Legal Information

Disclaimer

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only.

Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

Confidentiality Notice

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit consent by Critical Manufacturing.

Copyright Information

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

Trademark Information

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.