

Packing

Estimated time to read: 13 minutes

Overview

A **Package** is created from one or more **Products**. The **Product Type** of which must be **Packaging**. And although associated to a **Product**, a package hierarchy can have various **Materials**.

In Critical Manufacturing MES, packing is understood to be any type of box, package or container used for the purposes of transportation or shipping. Additionally, packing can take place at many levels as shown in the example below:

```

flowchart LR
  subgraph Product
    direction LR
    subgraph Packing["Protective Packaging for every level"]
      direction LR
      A1[Transportation Packaging] -.->|Pallet| N1[Level 4]
      A2[Warehouse Packaging] -.->|Corrugated Box| N2[Level 3]
      A3[Secondary Packaging] -.->|Box| N3[Level 2]
      A4[Primary Packaging] -.->|Plastic Bag| N4[Level 1]
    end
    A5[Finished Product Materials] -.-> Packing
  end
end

classDef mermaid_title color:#000, fill:#fafafa, stroke:#fafafa, stroke-width:0x, font-size:100%, font-weight:200;
classDef mermaid_start color:#000, fill:#fafafa, stroke:#fafafa, color:#fafafa, stroke-width:0x, font-size:100%, visibility: hidden;
classDef mermaid_businessdata color:#000, fill:#65CDE8, stroke:#65CDE8, stroke-width:0px, font-size:100%;
classDef mermaid_nonbusinessdata color:#000, fill:#B7DEE8, stroke:#B7DEE8, stroke-width:0px, font-size:100%;
classDef mermaid_entity color:#000, fill:#FB9F53, stroke:#FB9F53, stroke-width:0px, font-size:100%;
classDef mermaid_entitylinked color:#000, fill:#FCD5B5, stroke:#FCD5B5, stroke-width:0px, font-size:100%;
classDef mermaid_context color:#000, fill:#B9CDE5, stroke:#B9CDE5, stroke-width:0px, font-size:100%;
classDef mermaid_optional color:#000, fill:#B7DEE8, stroke:#65CDE8, stroke-width:1px, font-size:100%, stroke-dasharray: 5 5;
classDef mermaid_state color:#000, fill:#d7e4bd, stroke:#000, stroke-width:1px, font-size:100%, font-weight:300;
class Main mermaid_entity
class Start mermaid_start
class A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12 mermaid_state
class L1,L2,L3,L4,L5,L6 mermaid_state
class C1,C2,C3,C4,C5,C6 mermaid_context
class N1,N2,N3,N4,N5,N6 mermaid_nonbusinessdata

```

Note

This tutorial will focus on Level 1 and Level 2 packaging, but many more levels are possible in Critical Manufacturing MES.

Packing levels

Critical Manufacturing MES distinguishes between two levels of packing as shown in the table below:

Level	Description	Driven by	Comments
Level 1 Primary (Product)	Packaging of the main product, which is typically a Manufacturing Process . This includes all the Materials that make up an individual Product Package. The contents of Packages at this level consist always of Materials.	A <u>BOM</u> .	1. Requires only one Product of Product Type Packaging in the <u>BOM</u> . 2. Supports Trackable (has a unique Id) and Non-Trackable Packages.
Level 2 and Above Secondary, Tertiary, and above (Package)	Packing of packages of Level 1 or above into other packages of a higher level (multiple level packaging), which is typically a Logistics Process . The contents of Packages at this level consist always of other Packages.	A Step level configuration together with a Packing Context Smart Table.	1. Will resolve to one Product of Product Type Packaging per level. 2. Multiple level packing is possible. 3. Level 2 Packages are always Trackable.

Table: Different Packing levels

Primary Packing

Primary packing consists of creating the packages of the first level - Level 1 packages. These are the packages that directly contain the WIP.

Secondary and Above Packing

Secondary and above packing consists of packing packages of Level 2 or above in other packages. This operation is referred to as multiple level packing.

Package

The **Package** entity is part of the **Material Logistics** module, which must be properly licensed in the system, and it can be accessed from the **Business Data** menu of Critical Manufacturing MES.

Note

The **Package** entity will only contain information if a packing operation has been executed.

Product

A **Product** is where packing begins, and a **Product** is always associated to a **Material**.

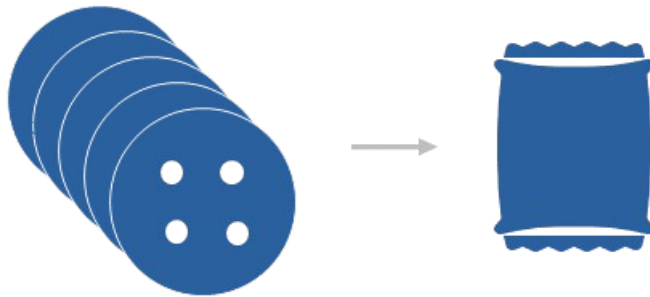
As every manufactured product needs to be shipped, it also needs to be packed. However, there are different packing processes depending on the immediate outcome of what is packed.

Critical Manufacturing MES caters for various packing processes and has related them to levels, as seen above.

In this tutorial we will understand how Level 1 and Level 2 are configured, and we will use a basic scenario with non-trackable packages for Level 1.

Packing Material - Level 1

Level 1 is also called product packing and it is typically created on the shop floor. Remember that the contents of packages at this level is always a **Material**. Therefore, for the purpose of this tutorial, for Level 1, we will pack cookies into individual plastic bags:



Setting up Packing Level 1

To have a functioning **Packing** module, you have to set up Critical Manufacturing MES entities as shown in the following table:

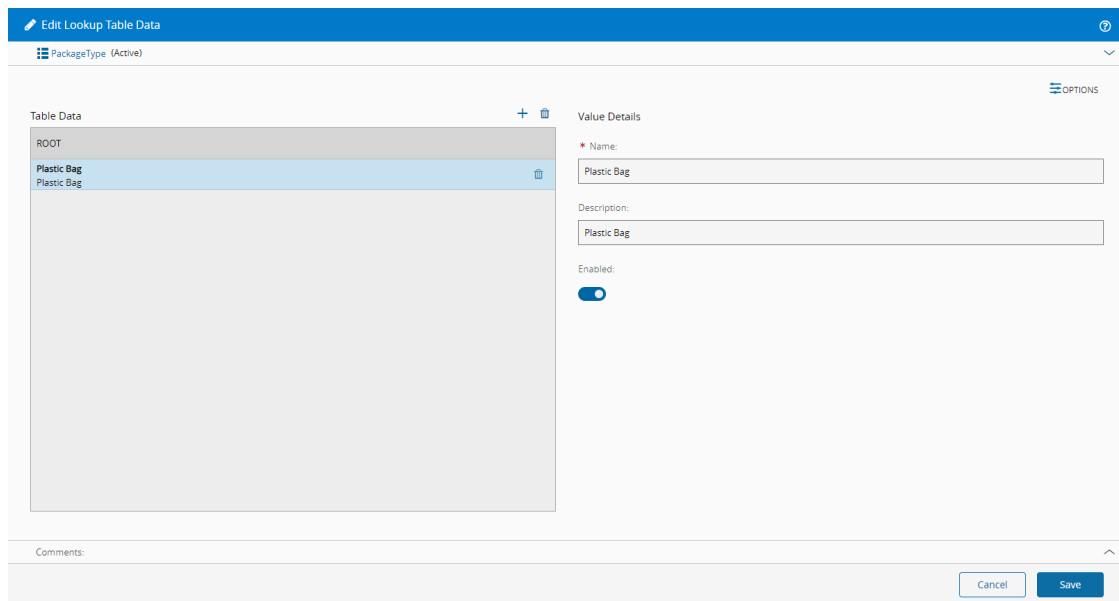
Step Number	Step	Description
1	Lookup Table	Edit the PackageType Lookup Table to create the needed package type.
2	Service	Create the needed Services .
3	Resource	Create the needed Resources .
4	Step	Create the needed Steps .
5	Flow	Create the needed Flows .
6	Product	Create the needed Products .
7	<u>BOM</u>	Create a <u>BOM</u> .
8	<u>BOM Context</u>	Define the <u>BOM Context</u> .

Table: Steps to set up Level 1 Packing

The next sub-sections will cover the required configuration steps in more detail.

Step 1: Lookup Table

1. Edit the **PackageType** Lookup Table to add the package type we will need. For Level 1 packing we will be using plastic bags:



Edit Lookup Table Data

PackageType (Active)

Table Data

ROOT
Plastic Bag
Plastic Bag

Value Details

* Name: Plastic Bag

Description: Plastic Bag

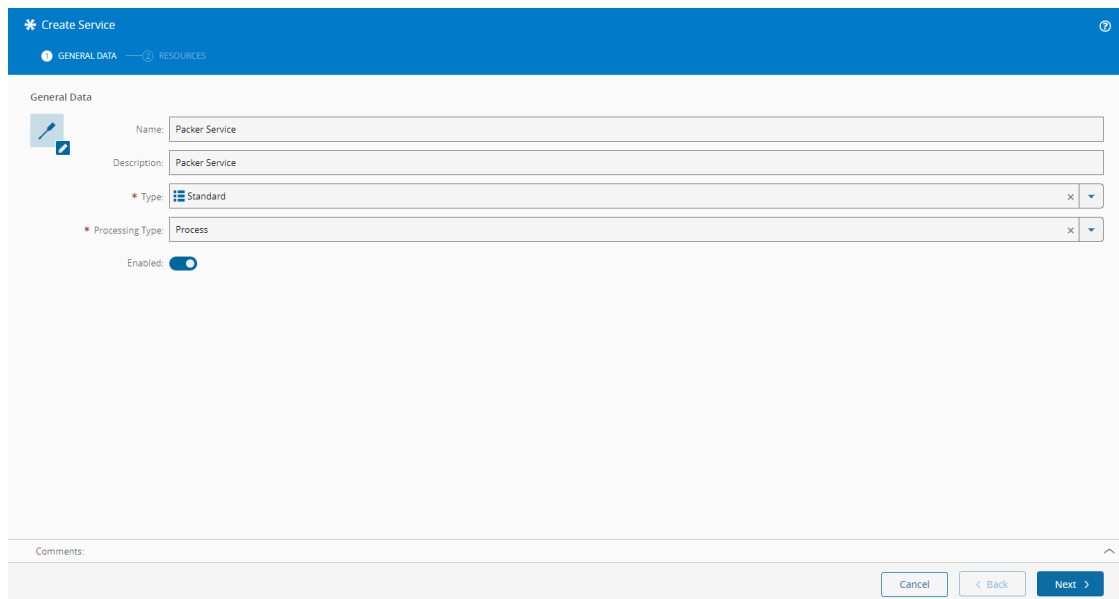
Enabled:

Comments:

Cancel Save

Step 2: Service

1. Create a **Service** of Processing Type **Process**:



Create Service

GENERAL DATA RESOURCES

General Data

Name: Packer Service

Description: Packer Service

* Type: Standard

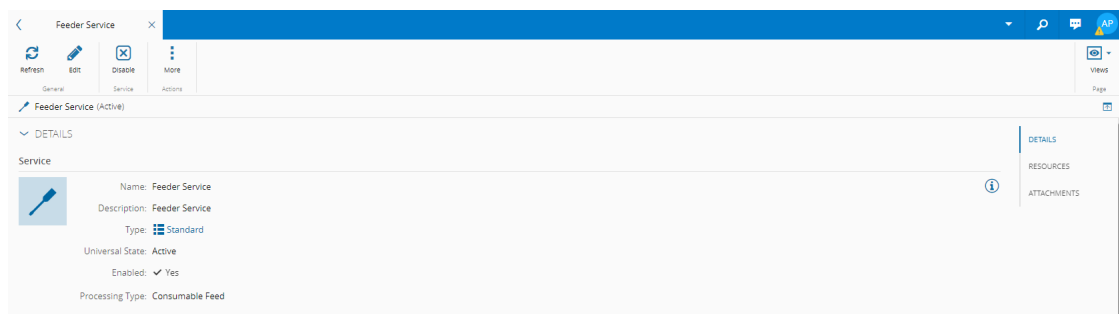
* Processing Type: Process

Enabled:

Comments:

Cancel < Back Next >

2. Create a **Service** of Processing Type **Consumable Feed**:



Feeder Service

Refresh Edit Disable More

Feeder Service (Active)

DETAILS

Service

Name: Feeder Service

Description: Feeder Service

Type: Standard

Universal State: Active

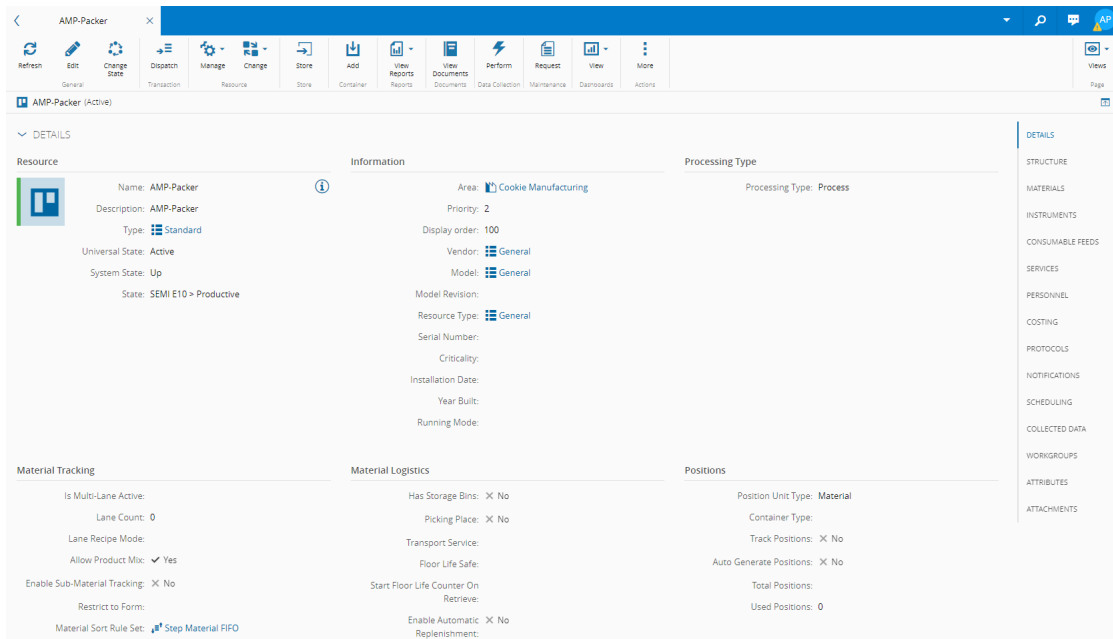
Enabled: Yes

Processing Type: Consumable Feed

DETAILS RESOURCES ATTACHMENTS

Step 3: Resource

1. Create a **Resource** of Processing Type **Process**. This is our main **Resource**, and we will configure it to have **Number of Consumable Feeds Positions** equal to 1:



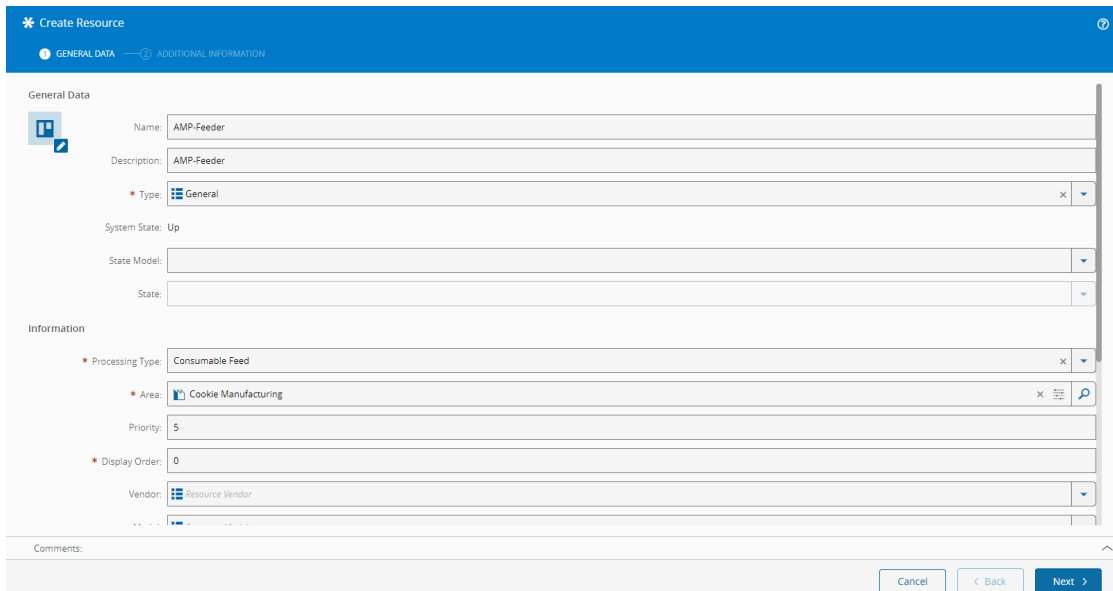
The screenshot shows the configuration page for the 'AMP-Packer' resource. The page is divided into several sections:

- Resource Information:** Name: AMP-Packer, Description: AMP-Packer, Type: Standard, Universal State: Active, System State: Up, State: SEMI E10 > Productive.
- Processing Type:** Processing Type: Process.
- Material Tracking:** Is Multi-Lane Active: No, Lane Count: 0, Lane Recipe Mode: No, Allow Product Mix: Yes, Enable Sub-Material Tracking: No, Restrict to Form: No, Material Sort Rule Set: Step Material RFO.
- Material Logistics:** Has Storage Bins: No, Picking Place: No, Transport Service: No, Floor Life Safe: No, Start Floor Life Counter On Retrieve: No, Enable Automatic Replenishment: No.
- Positions:** Position Unit Type: Material, Container Type: No, Track Positions: No, Auto Generate Positions: No, Total Positions: 0, Used Positions: 0.

Note

A **Resource** must have the same Number of Consumable Feeds Positions as **Products** in the **BOM**. For this tutorial we only need one.

2. Create a **Resource** of Processing Type **Consumable Feed**:



The screenshot shows the 'Create Resource' dialog box with the following configuration:

- General Data:** Name: AMP-Feeder, Description: AMP-Feeder, Type: General, System State: Up, State Model: (empty), State: (empty).
- Information:** Processing Type: Consumable Feed, Area: Cookie Manufacturing, Priority: 5, Display Order: 0, Vendor: Resource Vendor.

3. Manage Consumable Feeds of **Resource** of Processing Type **Process** to attach the **Resource** of Processing Type **Consumable Feed**:

Manage Consumable Feeds

AMP-Packer (Up) / SEMI E10 > Productive

Consumable Feeds (Used Positions 0/1)

POSITION	CONSUMABLE FEED
1	AMP-Feeder

Consumable Feed Details

Position: 1

Position Name:

Consumable Feed: AMP-Feeder

Comments:

Cancel Update

Note

Remember that **Resources** of Processing Type **Consumable Feed** are needed for the consumption of raw materials according to the **BOM**. Therefore, they need to be attached to the packing station where we are doing the packing.

4. Manage Services of **Resource** of Processing Type **Process** to add the previously created **Service** (Step 2 above):

AMP-Packer (Active)

SERVICES

Refresh Manage

Search

Services (1)

SERVICE	DESCRIPTION	TYPE	PRIORITY	ENABLED
Packer Service	Packer Service	Standard		✓

DETAILS
STRUCTURE
MATERIALS
INSTRUMENTS
CONSUMABLE FEEDS
SERVICES

5. Manage Services of **Resource** of Processing Type **Consumable Feed** to add the previously created **Service** (Step 2 above):

AMP-Feeder (Active)

SERVICES

Refresh Manage

Search

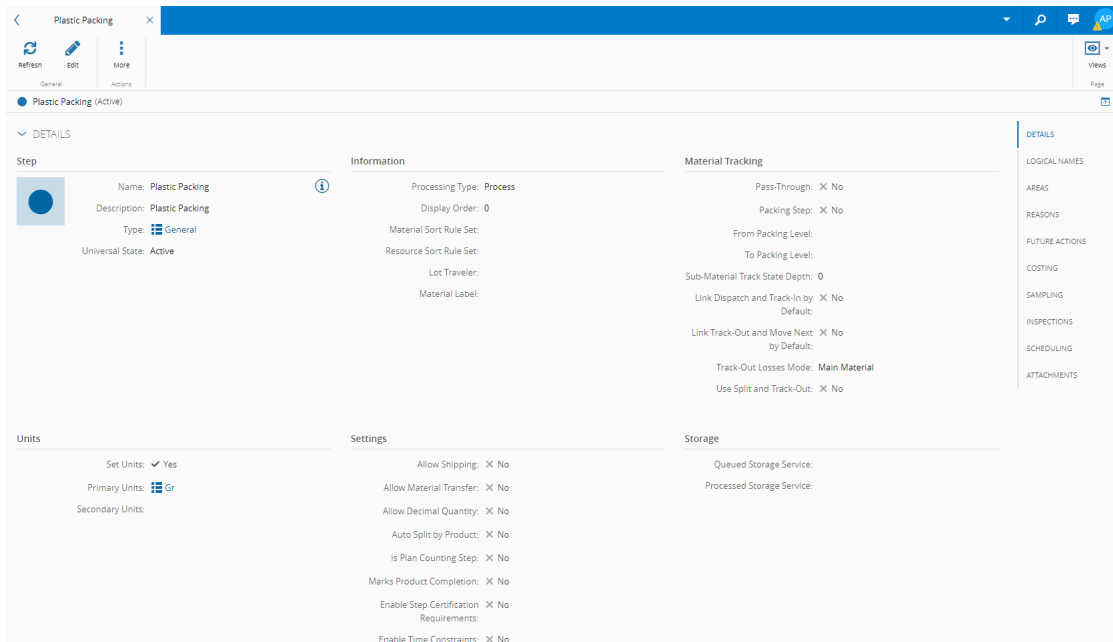
Services (1)

SERVICE	DESCRIPTION	TYPE	PRIORITY	ENABLED
Feeder Service	Feeder Service	Standard		✓

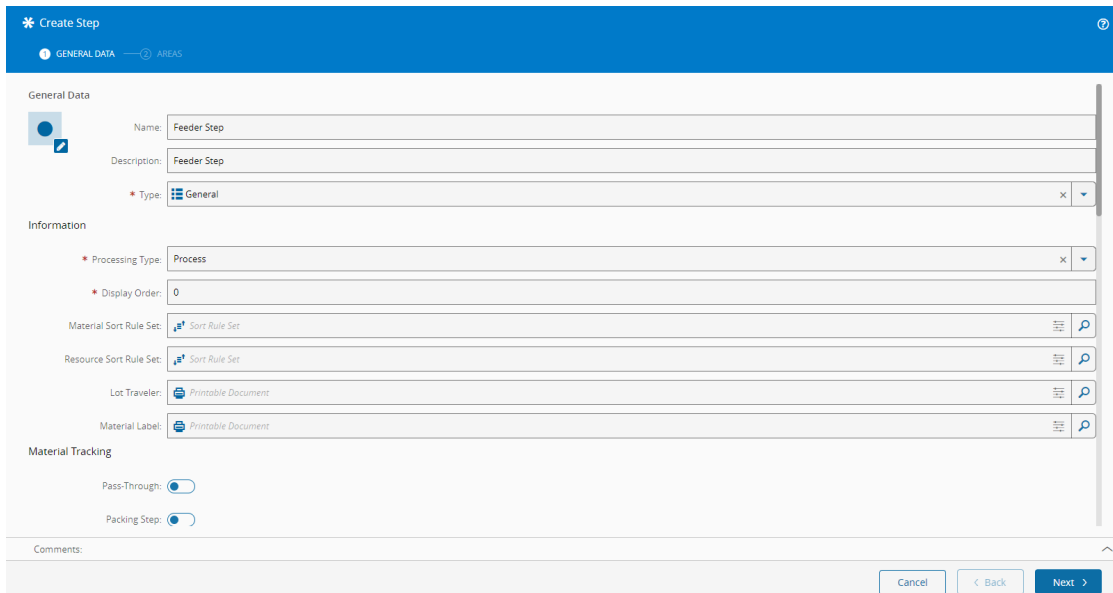
DETAILS
STRUCTURE
MATERIALS
SERVICES
PERSONNEL
COSTING

Step 4: Step

1. Create a **Step** of Processing Type **Process**. We will be using **Grams** as the Primary Units, and **Cookie Manufacturing** for the Area:



2. Create another **Step** with the same configurations as above but with a different name:



3. In the **Service Context** view of the **Steps** we just created, add the previously created **Services** (Step 2 above):

+ Add Service Context Record(s)

ServiceContext (Active)

Record(s)

Record #1
Step: Plastic Packing | Service: Packer Service

Service Context

Logical Flow Path:

Product:

Product Group:

Flow:

Material:

Material Type:

Production Order:

* Service:

Comments:

+ Add Service Context Record(s)

ServiceContext (Active)

Record(s)

Record #1
Step: Feeder Step | Service: Feeder Service

Service Context

Logical Flow Path:

Product:

Product Group:

Flow:

Material:

Material Type:

Production Order:

* Service:

Comments:

4. If we now **Refresh** the Service Context of the created **Services**, the corresponding **Steps** will be available:

Packer Service (Active)

SERVICE CONTEXT

Refresh

Step: Step Logical Flow Path: Product:

STEP	LOGICAL FLOW PATH	PRODUCT	PRODUCT GROUP	FLOW	MATERIAL	MATERIAL TYPE	PRODUCTION ORDER
<input type="checkbox"/>							
<input type="checkbox"/>							

Rows per Page: 25 Page 1 of 1 (1 Records)

Feeder Service (Active)

SERVICE CONTEXT

Refresh

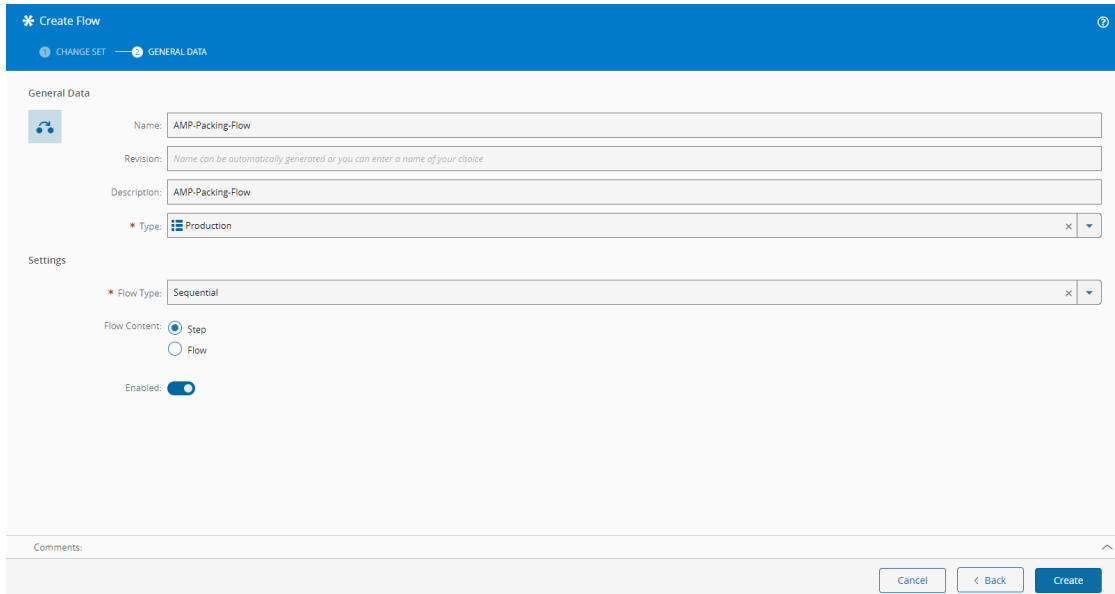
Step: Step Logical Flow Path: Product:

STEP	LOGICAL FLOW PATH	PRODUCT	PRODUCT GROUP	FLOW	MATERIAL	MATERIAL TYPE	PRODUCTION ORDER
<input type="checkbox"/>							
<input type="checkbox"/>							

Rows per Page: 50 Page 1 of 1 (1 Records)

Step 5: Flow

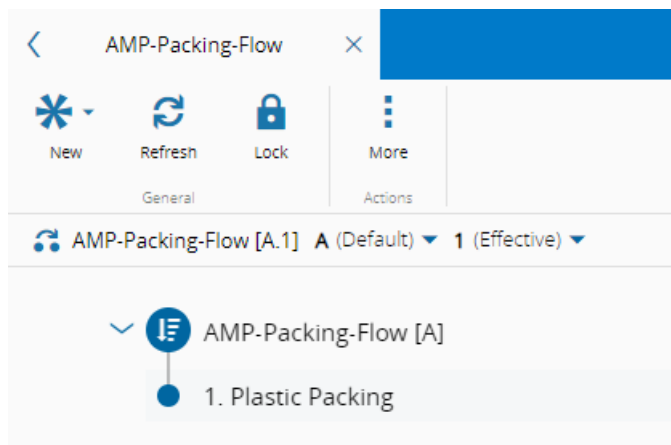
1. Create a **Flow**. For this tutorial we will use **Production** for the Type and **Sequential** for the Flow Type:



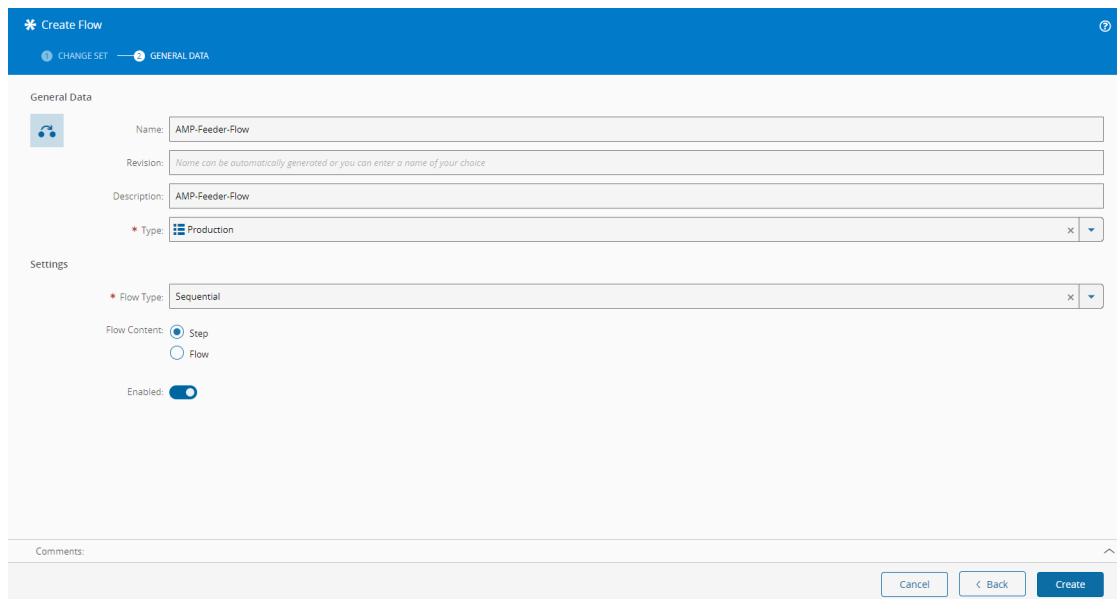
The screenshot shows the 'Create Flow' dialog box with the following fields and settings:

- General Data:**
 - Name: AMP-Packing-Flow
 - Revision: *Name can be automatically generated or you can enter a name of your choice*
 - Description: AMP-Packing-Flow
 - Type: Production
- Settings:**
 - Flow Type: Sequential
 - Flow Content: Step, Flow
 - Enabled:
- Comments:** (Empty text area)
- Buttons:** Cancel, < Back, Create

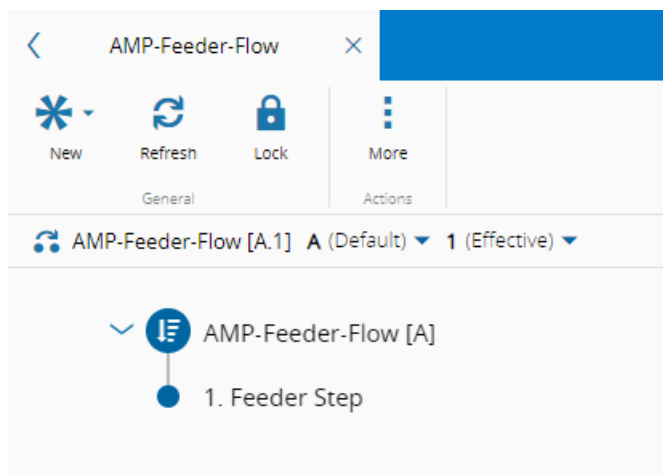
2. Edit the **Flow**, add the **Step** Plastic Packing, **Save**, and set Effective:



3. Create another **Flow** with the same configurations as above but with a different name:

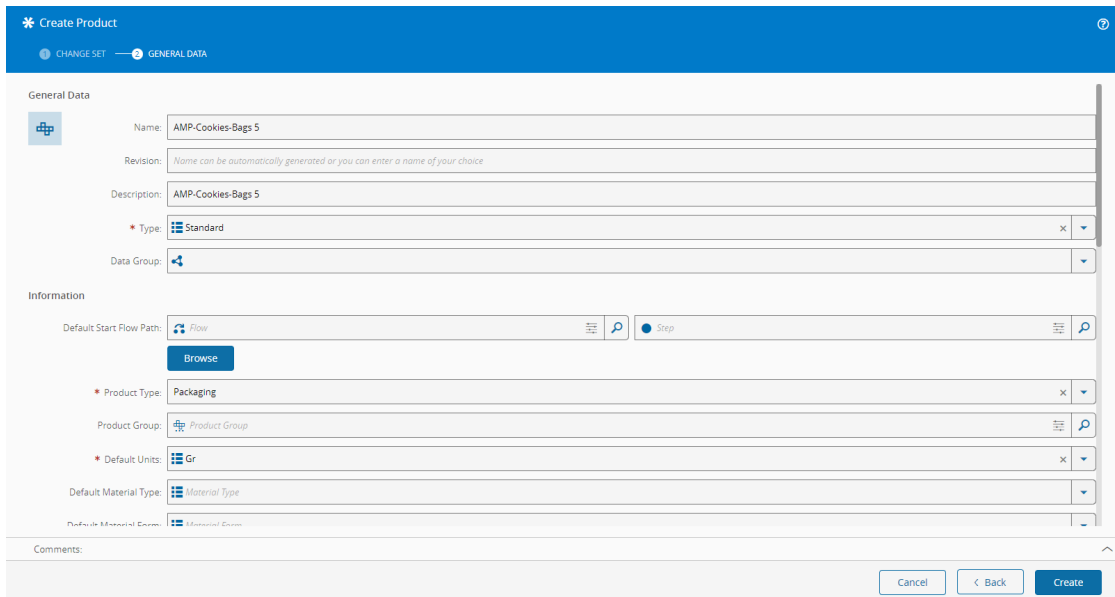


4. Edit the **Flow**, add the **Step** Feeder Step, **Save**, and set Effective:



Step 6: Product

1. Create a **Product** that must have **Packaging** as its **Product Type**, and remember that this is the **Product** we will use to pack our cookies - the one that will be in our **BOM**. The **Default Units** is Grams (**Units** Lookup Table):



Create Product

CHANGE SET — GENERAL DATA

General Data

Name: AMP-Cookies-Bags 5

Revision: Name can be automatically generated or you can enter a name of your choice

Description: AMP-Cookies-Bags 5

* Type: Standard

Data Group: [Select]

Information

Default Start Flow Path: [Flow] [Browse] [Step]

* Product Type: Packaging

Product Group: [Product Group]

* Default Units: Gr

Default Material Type: [Material Type]

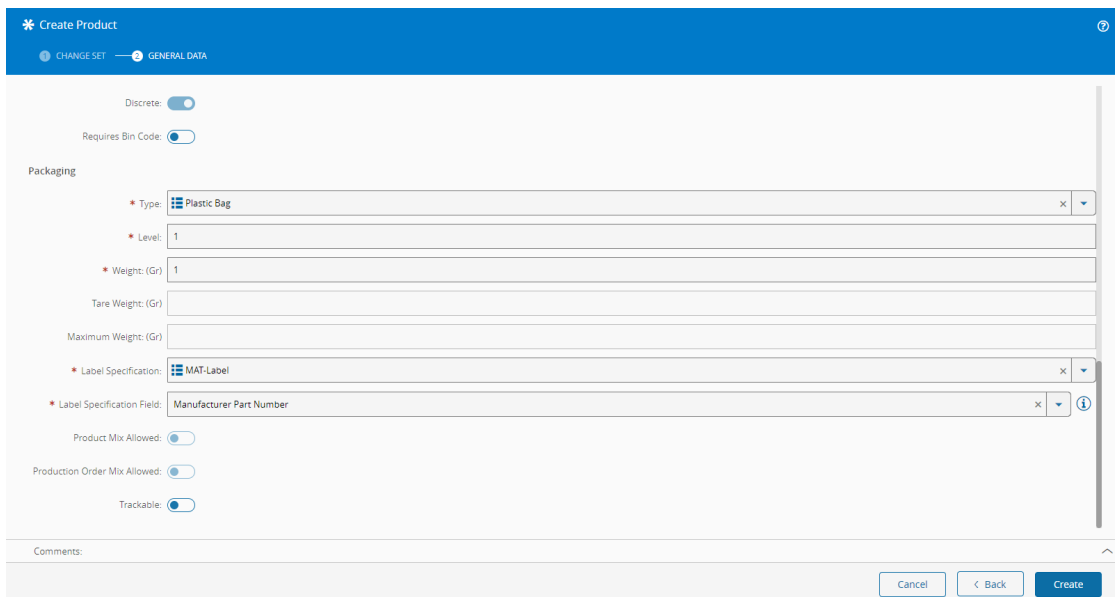
Default Material Form: [Material Form]

Comments:

Cancel < Back Create

In the **Packing** section of this wizard remember to enter the **Type** as previously defined in the **PackageType** Lookup Table, the **Level**, and the **Weight** (this weight refers to the plastic bag).

We will also have to enter **Label Specification** (**LabelSpecification** Generic Table) and **Label Specification Field** (**LabelSpecificationField** Generic Table) information. We need these fields because the packing process frequently uses barcode scanning, and we need to be able to read the package when, for example, we are adding a package of Level 1 into a package of Level 2.



Create Product

CHANGE SET — GENERAL DATA

Discrete:

Requires Bin Code:

Packaging

* Type: Plastic Bag

* Level: 1

* Weight: (Gr) 1

Tare Weight: (Gr)

Maximum Weight: (Gr)

* Label Specification: MAT-Label

* Label Specification Field: Manufacturer Part Number

Product Mix Allowed:

Production Order Mix Allowed:

Trackable:

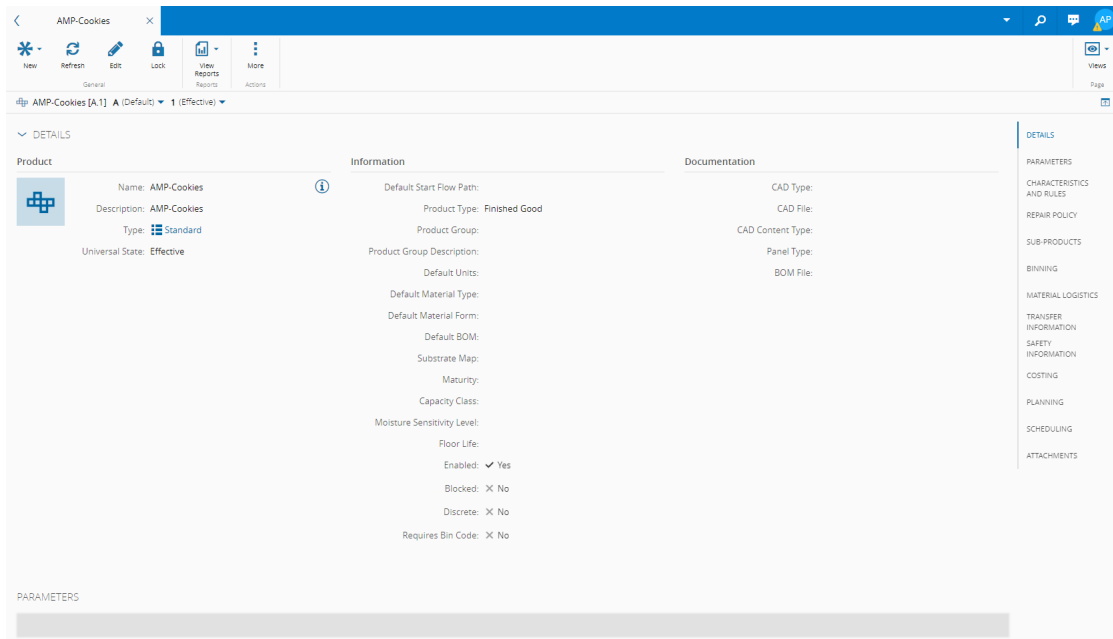
Comments:

Cancel < Back Create

Note

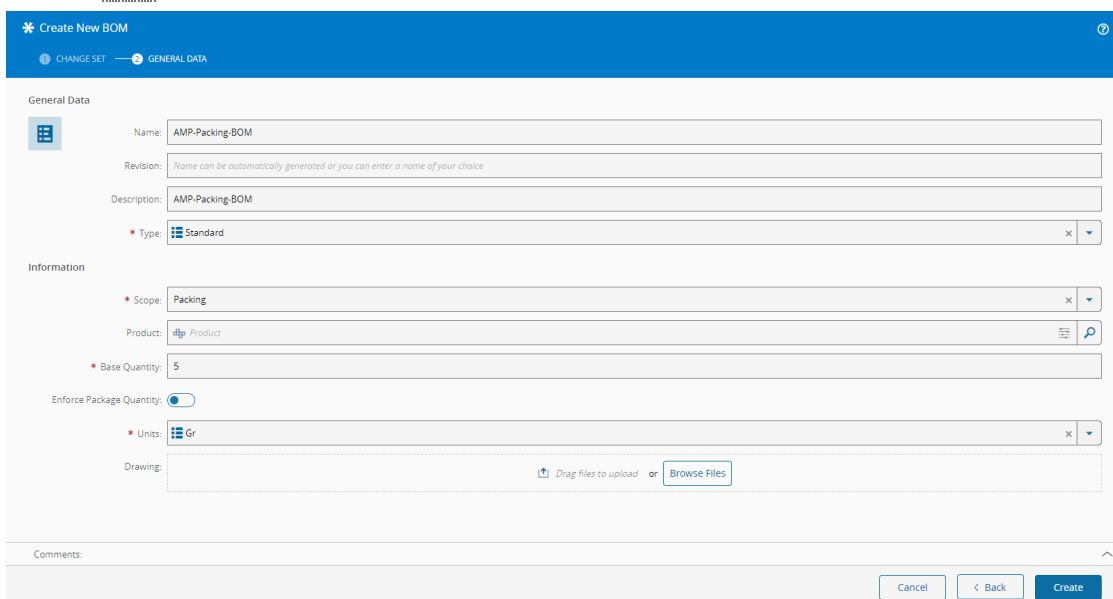
Remember that because we are not tracking Level 1 packages, the **Label Specification** defined here will then also be used for Level 2.

2. Create a **Product** of Type **Finished Good**. This will be part of our **Material** to pack:



Step 7: BOM

1. Create a **BOM** that must have **Packing** as its **Scope**, and the **Units** are as before:



A **BOM** of this scope must only contain one **Product** of **Packaging** type, which is used as the reference to calculate the contents of the plastic bags. We could have other **Products** as **BOM Items** in this **BOM**, such as stickers for the outside of the bags (informative or fun) or small gifts to go with the cookies in the bags, but their **Product Type** would have to be other than **Packaging**.

For the purpose of this tutorial our **BOM** will contain only the one necessary **Packaging** type **Product**.

2. Manage **BOM Items** and add the previously created **Product**:

AMP-Packing-BOM [A.1] A (Default) 1 (Created)

BOM ITEMS

Refresh Manage

Search

BOM Items (1)

ITEM	LOCATION	PRODUCT	PRODUCT DESCRIPTION	QUANTITY	UNITS	POSITION	SOURCE STEP	REFERENCE	ASSEMBLY STEP	LOGICAL FLOW PATH
1		AMP-Cookies-Bags 5 [A]	AMP-Cookies-Bags 5	1	Gr		Feeder Step	X	Plastic Packing	

Rows per Page: 25 Page 1 of 1 (1 Records)

Step 8: BOM Context

1. Configure the **BOM Context** in the packing step, which for this example is **Plastic Packing**:

+ Add BOM Context Record(s)

BOMContext (Active)

Record(s)

Record #1
Step: Plastic Packing | BOM: AMP-Packing-BOM | AssemblyType: Packing

BOM Context

Logical Flow Path: Browse

Product: Product

Product Group: Product Group

Flow: Flow

Material: Material

Production Order: Production Order

* BOM: AMP-Packing-BOM x

* Assembly Type: Packing x

Track-In Check Mode:

Track-Out Losses Mode:

Weigh and Dispense Mode:

Comments:

Cancel Add

2. We can also configure the **BOM Context** by using the **BOMContext** Smart Table. In this case, **Step** information has to be entered:

+ Add BOM Context Record(s)

BOMContext (Active)

Record(s)

Record #1
Step: Plastic Packing | BOM: AMP-Packing-BOM | AssemblyType: Packing

BOM Context

* Step: Plastic Packing x

Logical Flow Path: Browse

Product: Product

Product Group: Product Group

Flow: Flow

Material: Material

Production Order: Production Order

* BOM: AMP-Packing-BOM x

* Assembly Type: Packing x

Track-In Check Mode:

Track-Out Losses Mode:

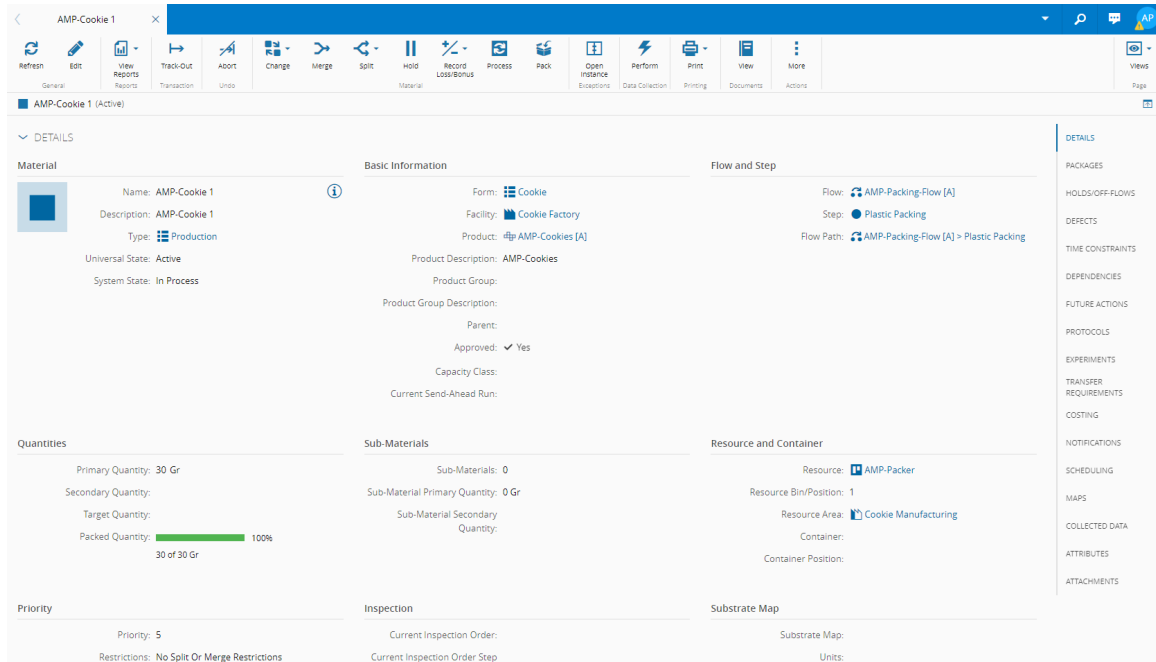
Weigh and Dispense Mode:

Comments:


Cancel Add

Level 1 Packing Test

To test that Level 1 packing is working we should create a **Material** with the cookies **Product**, we need to make sure we have created a **Material** containing the cookies bag **Product**, and it is attached to the feeder **Resource**, or we can attach this consumable **Material** inside the packing wizard using the Insert Barcode function. Then Dispatch and Track-In our cookie **Material**. At this point the **Pack** button should be available in the top ribbon:



We are now ready to pack our cookie **Material** following the instructions provided in [Pack Material](#).

 **Note**

In the image above we have already packed our cookie **Material** at Level 1. This is, the **Packed Quantity** is 100%, and we can see its details by selecting the **Packages** section of this page.

Packing Material - Level 2

Level 2 is also called logistics packing, and it relates to multiple levels of packing that is required for transportation and distribution. It consists of packing packages of Level 2 or above into other packages. Remember that the packages of Level 2 are always trackable, and that the contents of packages at this level are always other packages. Moreover, for the purpose of this tutorial, we will be packing our plastic bags, into a Level 2 box package:



Setting up Packing Level 2

To have a functioning **Packing** module, you have to set up Critical Manufacturing MES entities as shown in the following table:

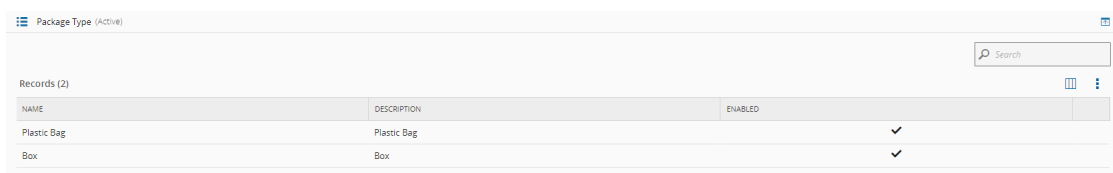
Step Number	Step	Description
1	Lookup Table	Edit the PackageType Lookup Table to create the needed package type.
2	Service	Create a Service .
3	Resource	Create a Resource .
4	Step	Create a Step .
5	Flow	Create version of previous Flow .
6	Product	Create a Product .
7	Smart Table	Edit the MaterialPackingContext Smart Table to create the needed package type.

Table: Steps to set up Level 2 Packing

The next sub-sections will cover the required configuration steps in more detail.

Step 1: Lookup Table

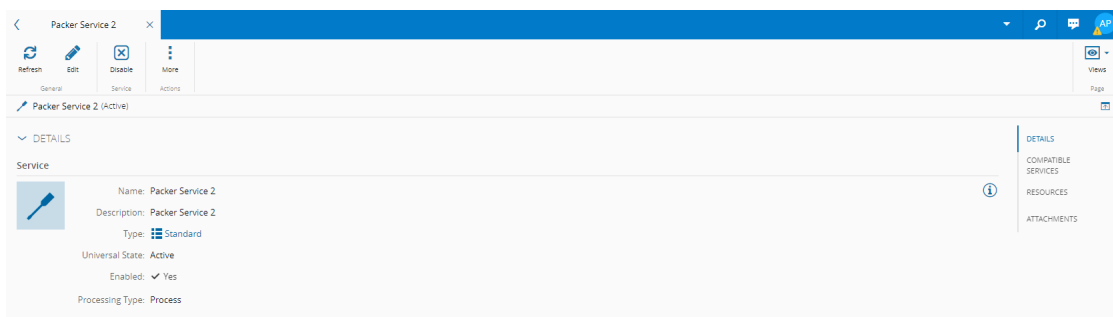
1. Edit the [PackageType](#) Lookup Table to add the package type we will need. For Level 2 packing we will be using boxes:



NAME	DESCRIPTION	ENABLED
Plastic Bag	Plastic Bag	✓
Box	Box	✓

Step 2: Service

1. Create a **Service** of Processing Type **Process**:

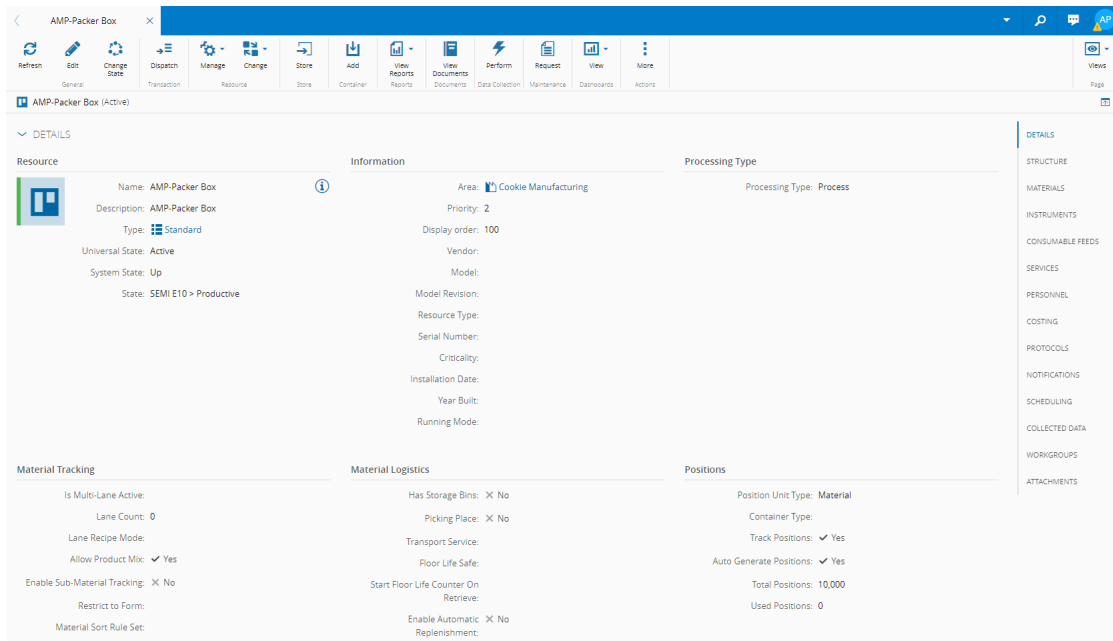


Service

- Name: Packer Service 2
- Description: Packer Service 2
- Type: Standard
- Universal State: Active
- Enabled: Yes
- Processing Type: Process

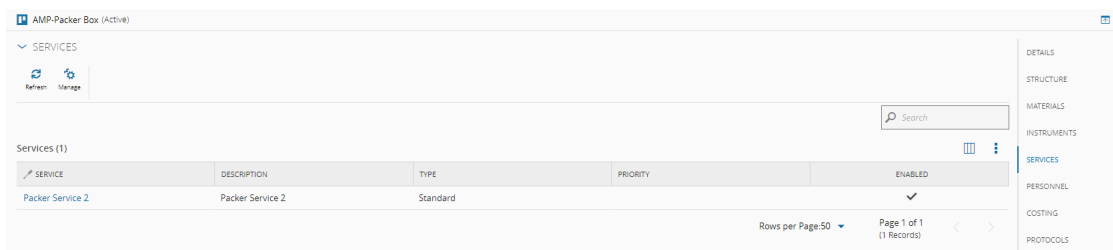
Step 3: Resource

1. Create a **Resource** of Processing Type **Process**:



The screenshot shows the SAP Resource configuration for 'AMP-Packer Box'. The 'Processing Type' is set to 'Process'. The 'Area' is 'Cookie Manufacturing'. The 'Priority' is 2, and the 'Display order' is 100. The 'Universal State' is 'Active', and the 'System State' is 'Up'. The 'State' is 'SEMI E10 > Productive'. The 'Material Tracking' section includes 'Is Multi-Lane Active' (unchecked), 'Lane Count' (0), 'Lane Recipe Mode' (unchecked), 'Allow Product Mix' (checked), 'Enable Sub-Material Tracking' (unchecked), and 'Restrict to Form' (unchecked). The 'Material Logistics' section includes 'Has Storage Bins' (unchecked), 'Picking Place' (unchecked), 'Transport Service' (unchecked), 'Floor Life Safe' (unchecked), 'Start Floor Life Counter On Retrieve' (unchecked), 'Enable Automatic Replenishment' (unchecked), and 'Year Built' (empty). The 'Positions' section includes 'Position Unit Type' (Material), 'Container Type' (unchecked), 'Track Positions' (checked), 'Auto Generate Positions' (checked), 'Total Positions' (10,000), and 'Used Positions' (0).

2. Associate the **Service** we just created (Step 2 above) to this **Resource**.



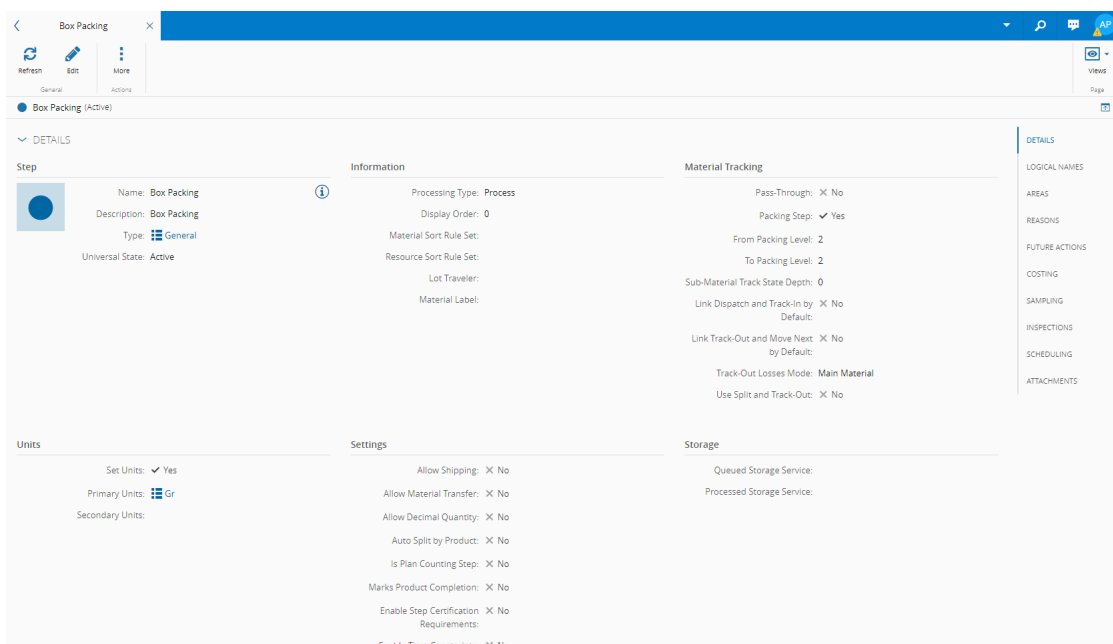
The screenshot shows the 'SERVICES' section for the 'AMP-Packer Box' resource. A table lists the associated services:

SERVICE	DESCRIPTION	TYPE	PRIORITY	ENABLED
Packer Service 2	Packer Service 2	Standard		<input checked="" type="checkbox"/>

The table shows 1 record. The page is 1 of 1 (1 Records). Rows per page: 50.

Step 4: Step

1. Create a **Step** of Processing Type **Process**. At this level we have to mark the **Packing Step** field as **True** and define the **From Packing Level** and **To Packing Level**:



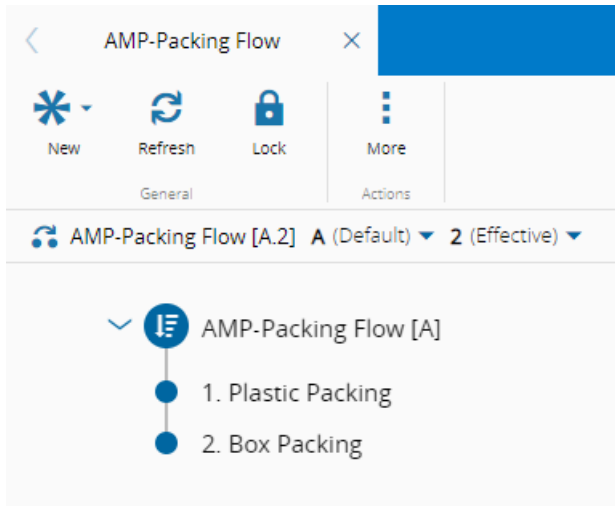
The screenshot shows the SAP Step configuration for 'Box Packing'. The 'Processing Type' is 'Process'. The 'Display Order' is 0. The 'Material Sort Rule Set' and 'Resource Sort Rule Set' are empty. The 'Lot Traveler' and 'Material Label' are empty. The 'Material Tracking' section includes 'Pass-Through' (unchecked), 'Packing Step' (checked), 'From Packing Level' (2), 'To Packing Level' (2), 'Sub-Material Track State Depth' (0), 'Link Dispatch and Track-In by Default' (unchecked), 'Link Track-Out and Move Next by Default' (unchecked), 'Track-Out Losses Mode' (Main Material), and 'Use Split and Track-Out' (unchecked). The 'Settings' section includes 'Allow Shipping' (unchecked), 'Allow Material Transfer' (unchecked), 'Allow Decimal Quantity' (unchecked), 'Auto Split by Product' (unchecked), 'Is Plan Counting Step' (unchecked), 'Marks Product Completion' (unchecked), 'Enable Step Certification Requirements' (unchecked), and 'Enable Time Constraints' (unchecked). The 'Storage' section includes 'Queued Storage Service' and 'Processed Storage Service' (both empty). The 'Units' section includes 'Set Units' (checked), 'Primary Units' (Gr), and 'Secondary Units' (empty).

Note

The **From Packing Level** and the **To Packing Level** must be defined with a value greater than 1. The **To Packing Level** must be greater or equal to the **From Packing Level**. For the purpose of this tutorial these values will be the same.

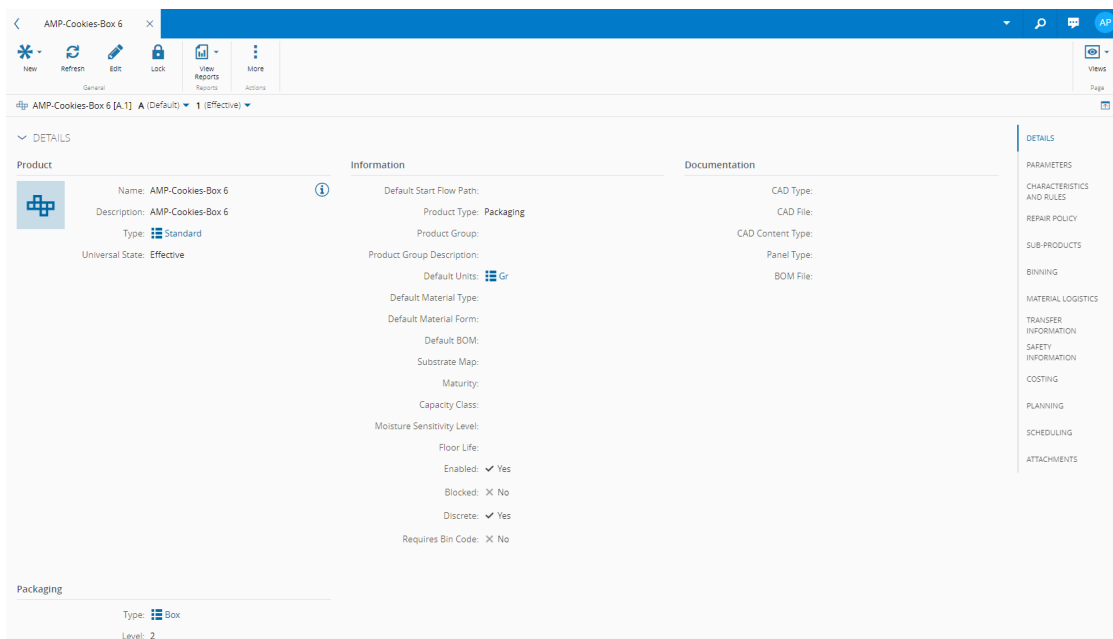
Step 5: Flow

1. Use the **Flow** created for Level 1, create a new **Version**, add the Box Packing **Step, Save**, and set Effective:



Step 6: Product

1. Create a **Product** that must have **Packaging** as its **Product Type**, and remember that this is the **Product** we will use to pack our cookie packs:

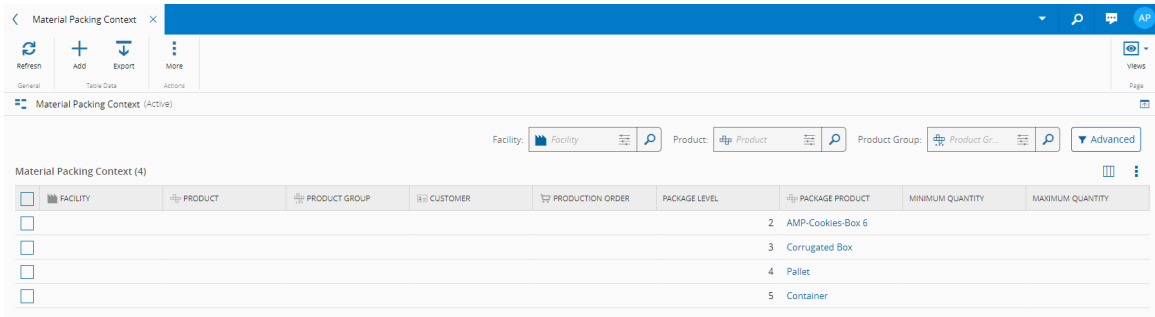


Note

We can already see that Level 2 is a hierarchy of packages (multiple level packing) where we pack smaller packages into larger packages, and we can do this at various levels, whereas Level 1 is a simple packing operation.

Step 7: Smart Table

1. Level 2 uses a Step Level Configuration (Step 4 above).
2. Configure the **MaterialPackingContext** Smart Table to establish the **Package Product** that will tell us for a certain material context and packing level what is the specific box/package product to be used:



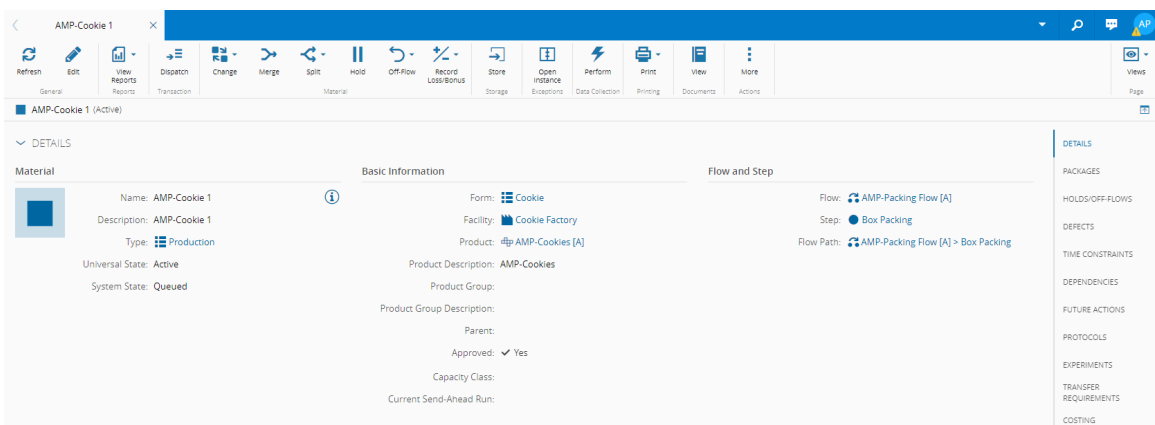
FACILITY	PRODUCT	PRODUCT GROUP	CUSTOMER	PRODUCTION ORDER	PACKAGE LEVEL	PACKAGE PRODUCT	MINIMUM QUANTITY	MAXIMUM QUANTITY
					2	AMP-Cookies-Box 6		
					3	Corrugated Box		
					4	Pallet		
					5	Container		

Note

For this tutorial we will only use **Package Level 2**, but the image above provides examples of other possible levels. Moreover, in Level 1, we know what package to use because it is part of the **BOM**. For Level 2 and above we resolve the package to use by configuring the **MaterialPackingContext** Smart Table.

Level 2 Packing Test

To test that Level 2 packing is working, we can use the **Material** we packed in Level 1. Track out the Level 1 **Material** and Move-Next to the Box Packing **Step**:



AMP-Cookie 1 (Active)

DETAILS

Material

Name: AMP-Cookie 1
 Description: AMP-Cookie 1
 Type: Production
 Universal State: Active
 System State: Queued

Basic Information

Form: Cookie
 Facility: Cookie Factory
 Product: AMP-Cookies [A]
 Product Description: AMP-Cookies
 Product Group: AMP-Cookies
 Product Group Description:
 Parent:
 Approved: Yes
 Capacity Class:
 Current Send-Ahead Run:

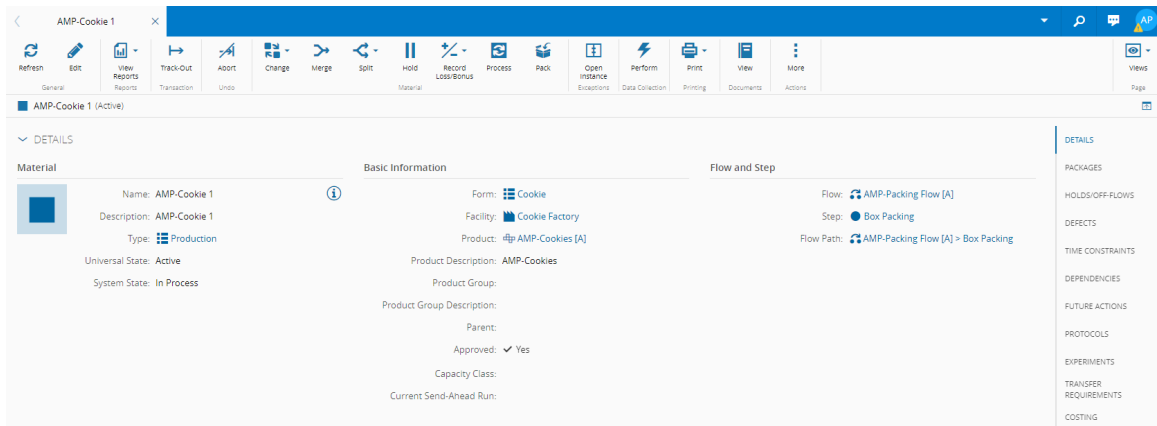
Flow and Step

Flow: AMP-Packing Flow [A]
 Step: Box Packing
 Flow Path: AMP-Packing Flow [A] > Box Packing

DETAILS

- PACKAGES
- HOLDS/OFF-FLOWS
- DEFECTS
- TIME CONSTRAINTS
- DEPENDENCIES
- FUTURE ACTIONS
- PROTOCOLS
- EXPERIMENTS
- TRANSFER REQUIREMENTS
- COSTING

Then Dispatch and Track-In the **Material** at which point the **Pack** button is available in the top ribbon:



We are now ready to pack our cookie **Material** following the instructions provided in [Pack Material Packages](#).