Critical manufacturing
an ASM PT company

Containers

11.2

March 2026

DOCUMENT ACCESS

Public

Containers

Estimated time to read: 9 minutes

Overview

A **Container** is a standardized physical carrier used to hold, protect, organize, and move **Materials**, parts, or other containers throughout manufacturing and logistics processes.

A Container acts as the interface between production, storage, and transportation. It ensures that items can be handled safely, efficiently, and repeatedly across the shop floor.

Manufacturers design containers to match:

- The size and weight of the contents
- The sensitivity of the Materials
- The handling methods used on the shop floor, such as manual handling, forklifts, conveyors, AGVs, or robots

Industrial containers can be:

- Returnable or disposable
- Standard or custom-designed
- Made from plastic, metal, wood, or composite materials

Beyond simple storage, containers support lean manufacturing principles by enabling:

- Visual management
- Work-in-process control
- Ergonomic handling
- Quality protection
- Traceability

A container is therefore not just a box. It is a critical logistics and production asset that connects material flow, safety, efficiency, and quality across the industrial shop floor.

Examples

FOUP for Semiconductor Wafers	**SMT Board Magazine**

Pallet

Trolley of Boxes

SMT Feeders Cart

ESD Box

Tray	Hopper
	

Containers in MES

A **Container** is an object of the **Container** entity that can hold **Materials**, **Resources**, or other **Containers**.

A **Material Container** has a capacity defined by positions available for a given **Material** unit. A **Container** can hold multiple **Materials**, but each position can contain only one **Material**.

Each position:

- Is identified by a unique number
- Has a defined capacity based on specific units

The same concept applies to:

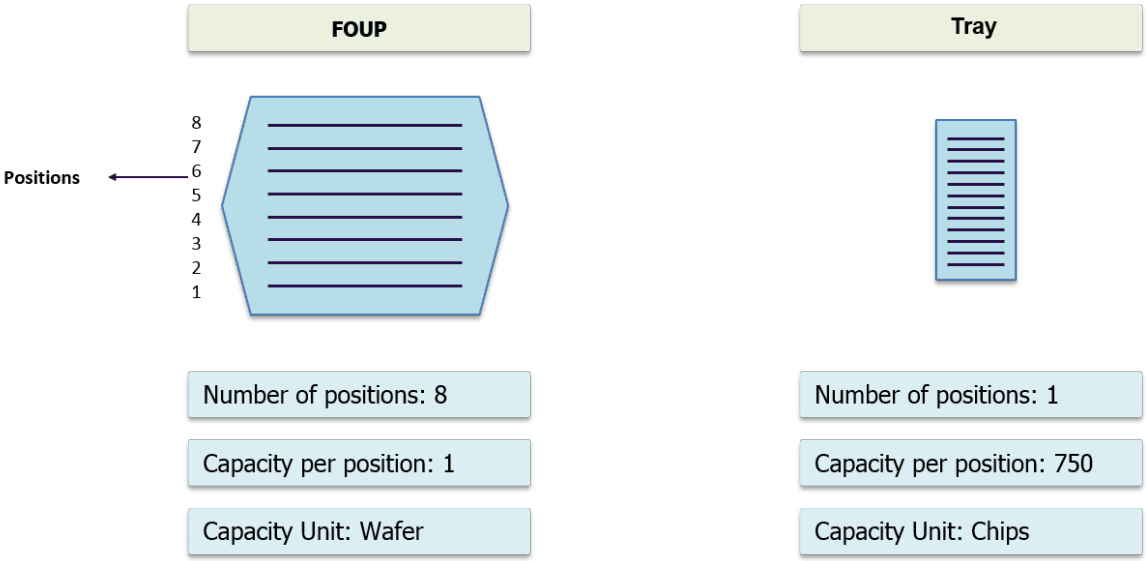
- **Containers** that hold **Resources**
- **Containers** that hold other **Containers** (SubContainers)

```
flowchart TD

    SC[Sub Container]
    C[Container]

    F[Facility]
    MC[Material Container]
    CR[Container Resource]

    M[Material]
    R[Resource]

    SC --> C

    C --> F
    C --> MC
    C --> CR

    MC --> M
    CR --> R

    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;
    class C,SC mermaid_entity
    class F,M,P,R mermaid_businessdata
    class MC,CR mermaid_nonbusinessdata
```

A **Container** can be modeled in different ways:

- **Magazine** — one position with multiple units per position (for example, a tray holding several units of the same Material)
- **FOUP (Front-Opening Universal Pod)** — multiple positions with one unit per position (for example, a wafer carrier)
- **Hybrid models** — combinations of both approaches



A **Container** can also be associated with a **Resource** when the Resource supports the position unit types defined in that Container type.

State Model

Containers have a system-defined state model:

- **Available** — the Container can be used (empty or full)
- **Unavailable** — the Container cannot be used
- **In Transit** — the Container is shipping Materials to another Facility

```
graph TD
    Start -->|Create Container| A1
    A1[Available] -->|Make Container Unavailable| A2[Unavailable]
    A2 -->|Make Container Available| A1
    A1 -->|Ship Material| L1[In Transit]
    L1 --->|Unship Material<br>Receive Material| A1
```

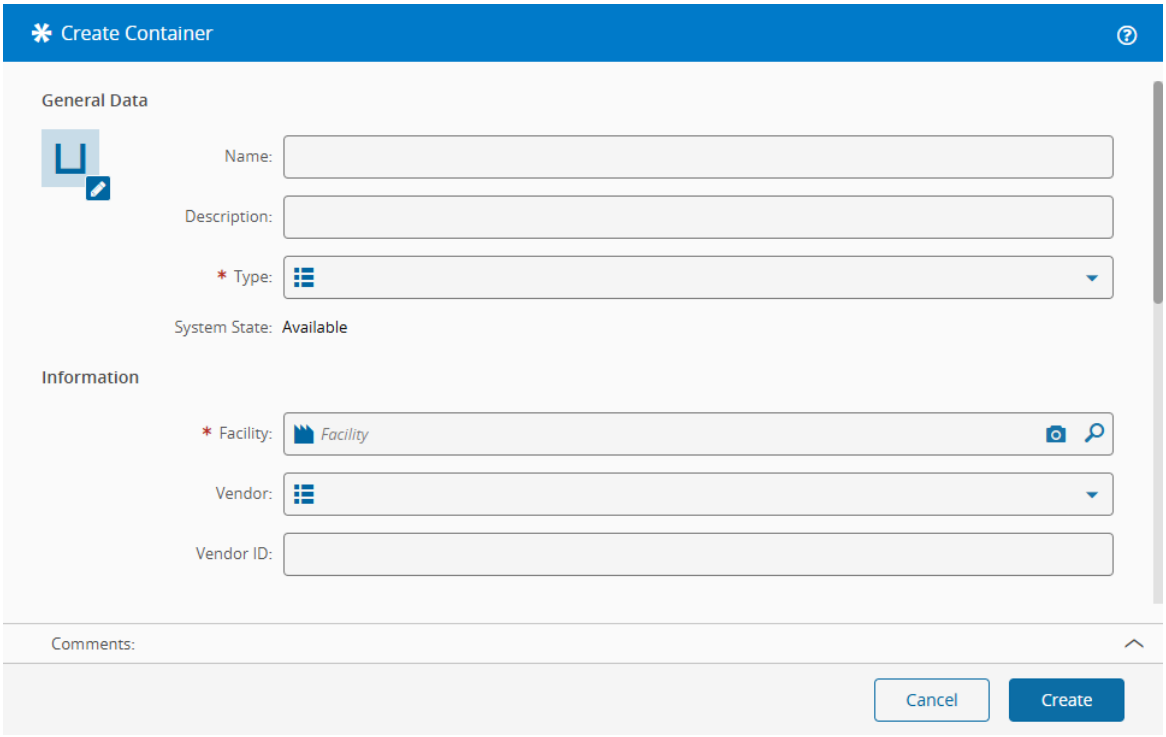
```
classDef mermaid_start color:#000, fill:#fafafa, stroke:#fafafa, color:#fafafa, stroke-width:0x,
font-size:100%, visibility: hidden;
classDef mermaid_entitylinked color:#000, fill:#FCD5B5, stroke:#FCD5B5, stroke-width:0px, font-
size:100%;
classDef mermaid_state color:#000, fill:#d7e4bd, stroke:#000, stroke-width:1px, font-size:100%,
font-weight:300;
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_entitylinked
```

You can also create custom **State Models** for Containers.

Container Creation

The Container creation wizard includes several required fields:

- **Name** — identifies the Container
- **Type** — used in configuration tables for container restrictions
- **Facility** — defines where the Container will be used

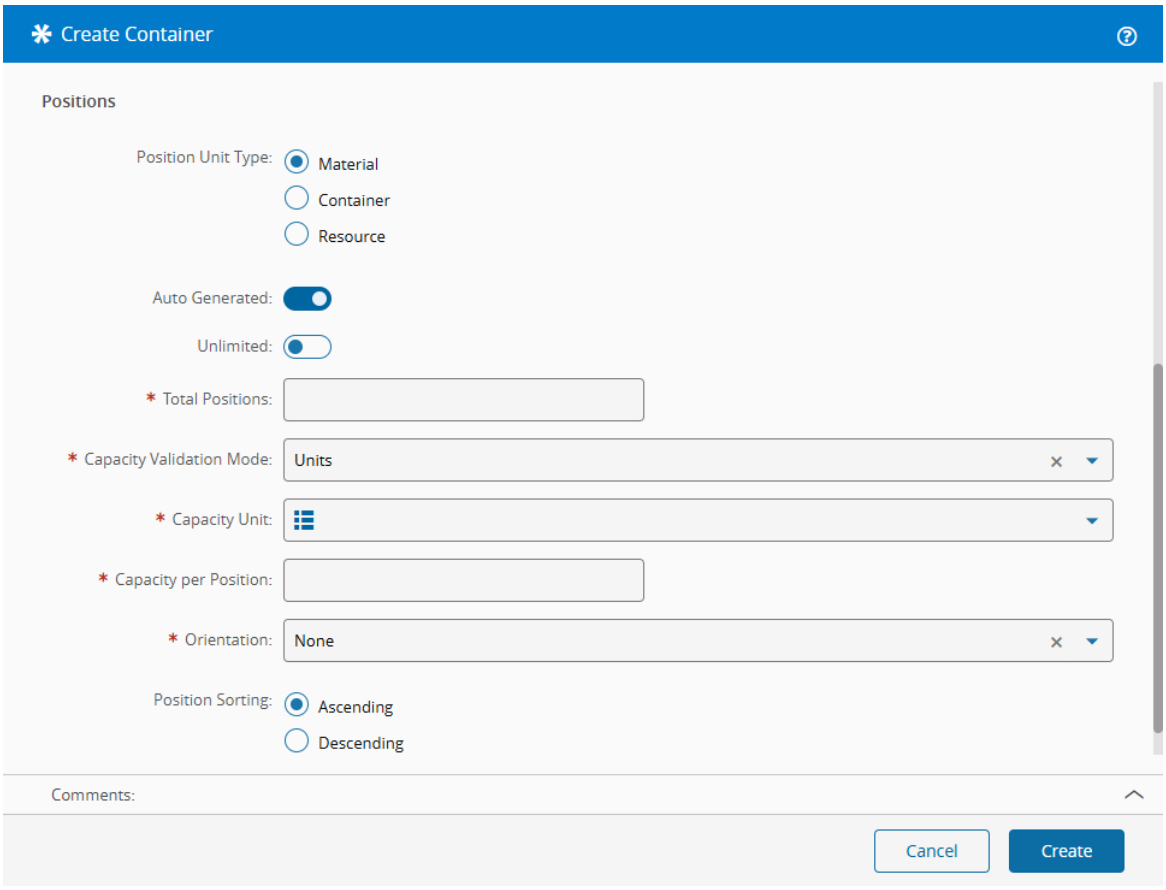


Additional configuration fields include:

- **Position Unit Type** — defines whether the Container holds Materials, Containers, or Resources
- **Total Positions** — defines the number of usable positions
- **Capacity Validation Mode** — available only for Material containers

If the **Capacity Validation Mode** is set to **Unit**, you can define the maximum Material quantity per position using **Capacity per Position**.

> **Info**
>
> For more information, refer to the User Guide: Create Container



Container Operations

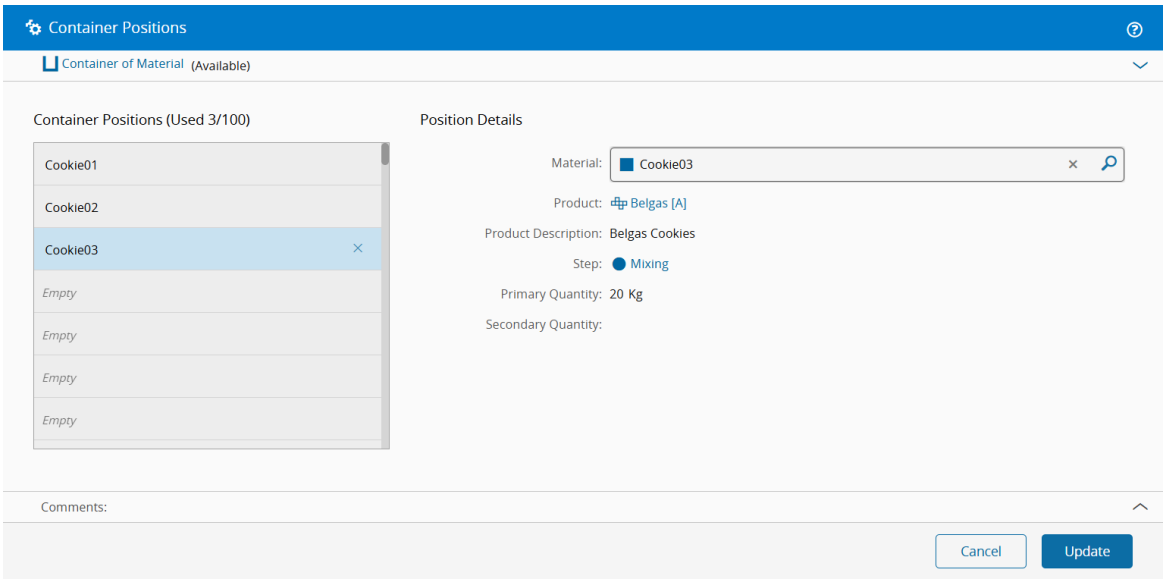
Manage Positions

Use **Manage Positions** to add or remove objects from Container positions.

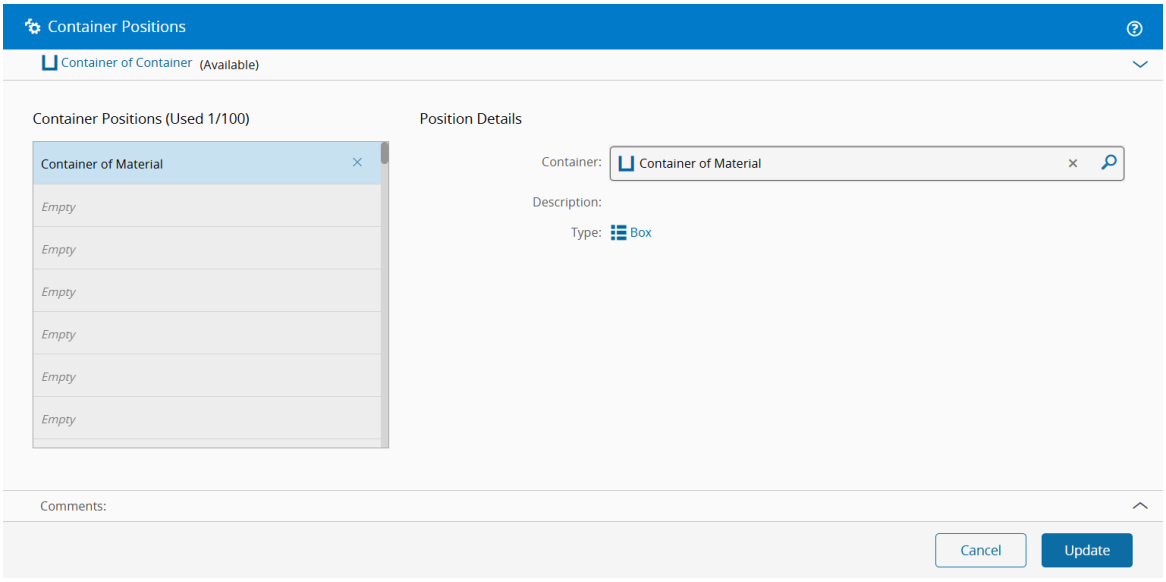
The operation works the same way for all **Container Position Unit Types**:

- Material
- Container
- Resource

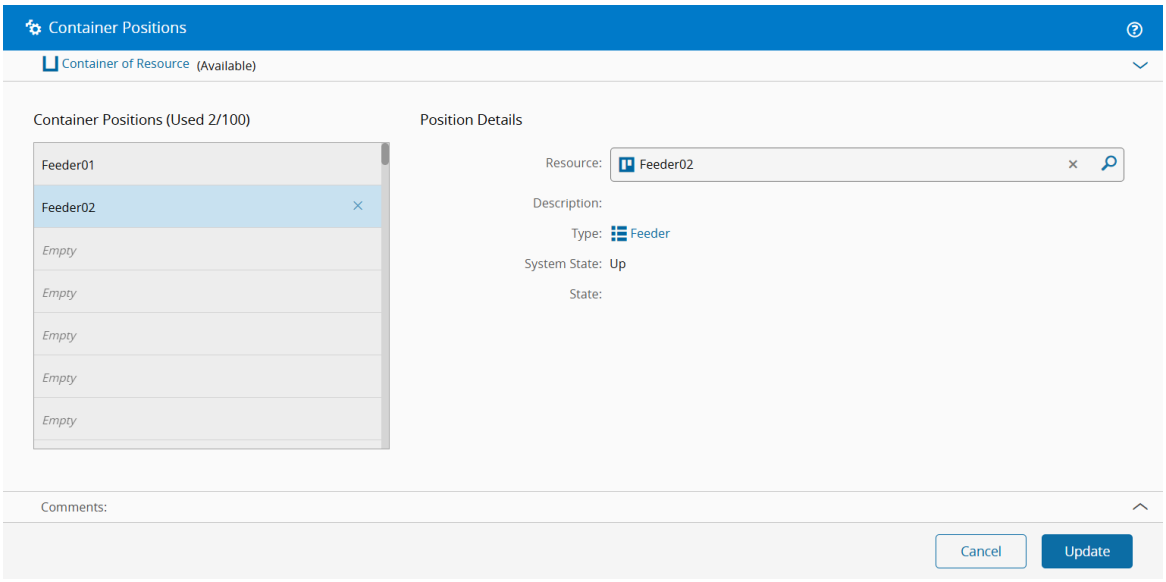
For **Material** or **Container**:



For **Container of Containers**, define the relationship between the container and sub-container in the **Container Type Relation** Generic Table.



For **Container of Resources**, define the relationship in the **Container Resource Relation** Generic Table.



Empty

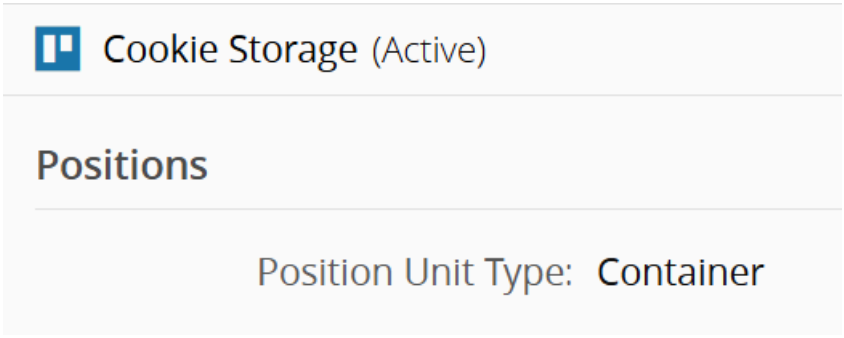
The **Empty** operation removes all objects from the Container.

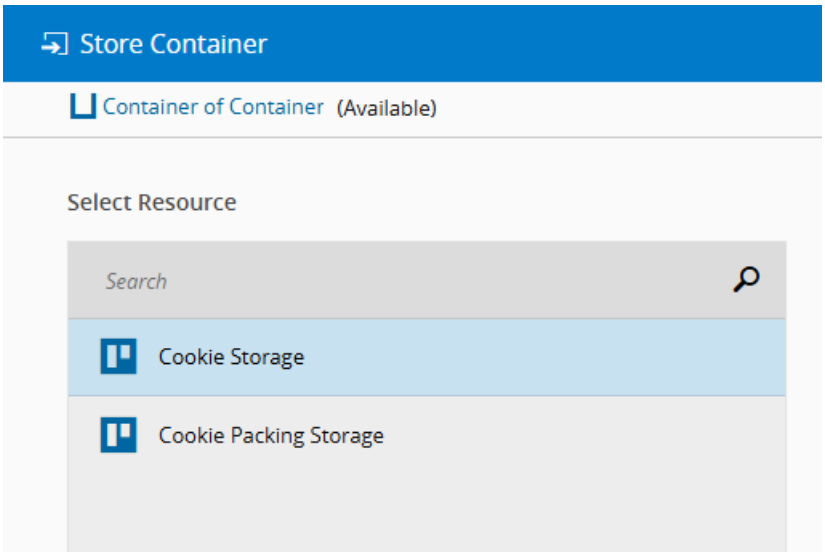
Store

The **Store** operation places the Container in a **Storage Resource** located in any **Area** that belongs to the Container **Facility**.

Precondition:

The Storage Resource must have **Position Unit Type = Container**.





Retrieve

The **Retrieve** operation removes the Container from storage.

Dock

Use **Dock** to dock a Container to a **Resource**.

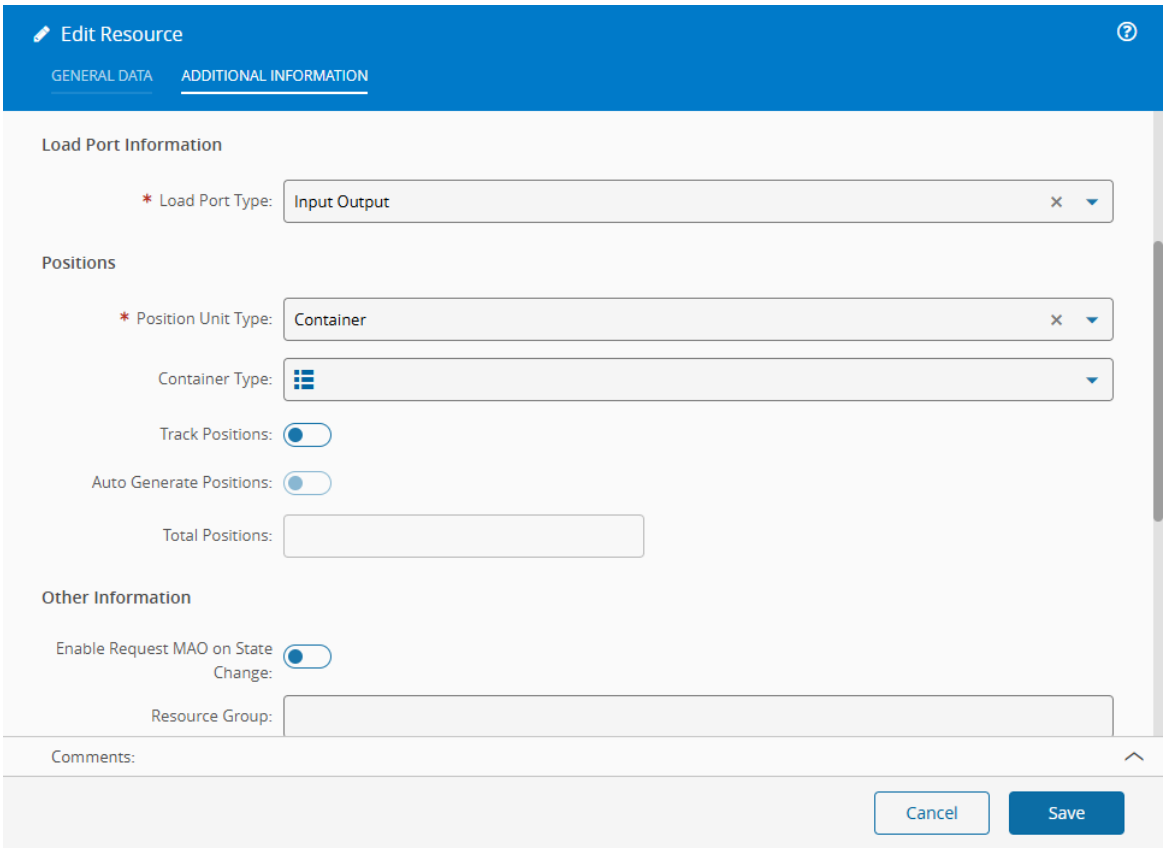
Docking can occur on Resources with the following processing types:

- **Load Port**
- **Process**

Load Port Docking

Requirements:

- The Resource must have a **Load Port Type** selected
- The **Position Unit Type** must be **Container**



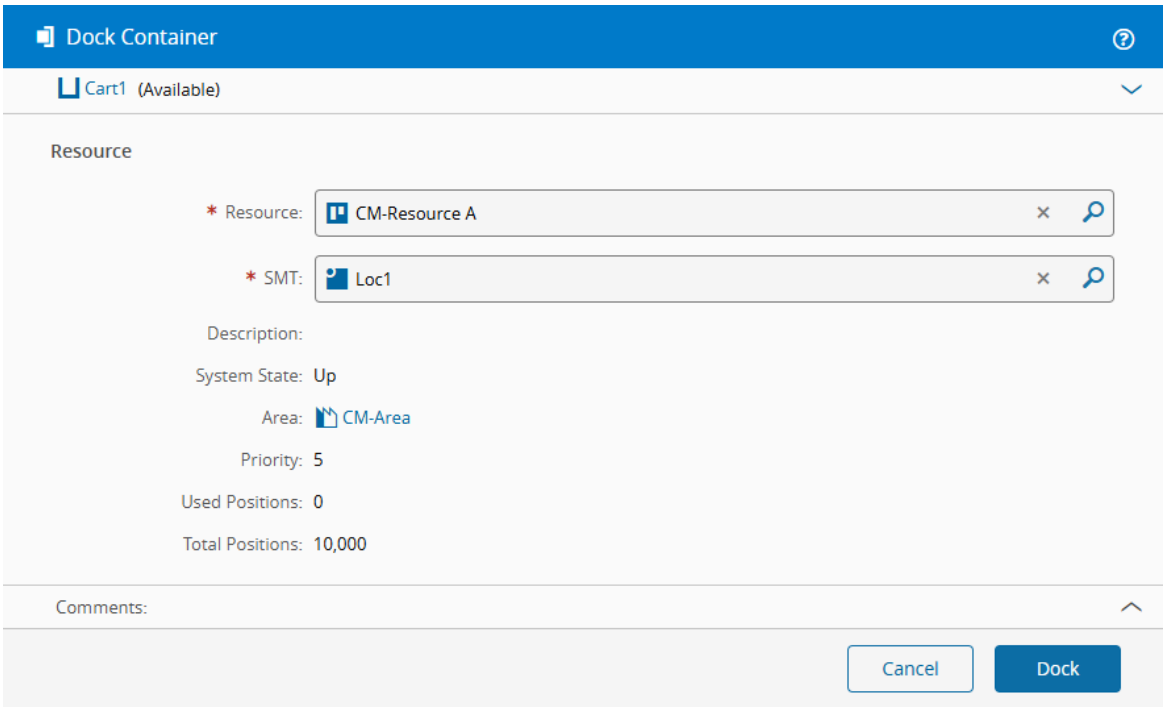
Example:

In SMT lines, loader and unloader machines handle magazines of electronic boards.

Process Docking

Requirements:

- The Resource must have **Locations enabled**
- A valid **Location** must be selected



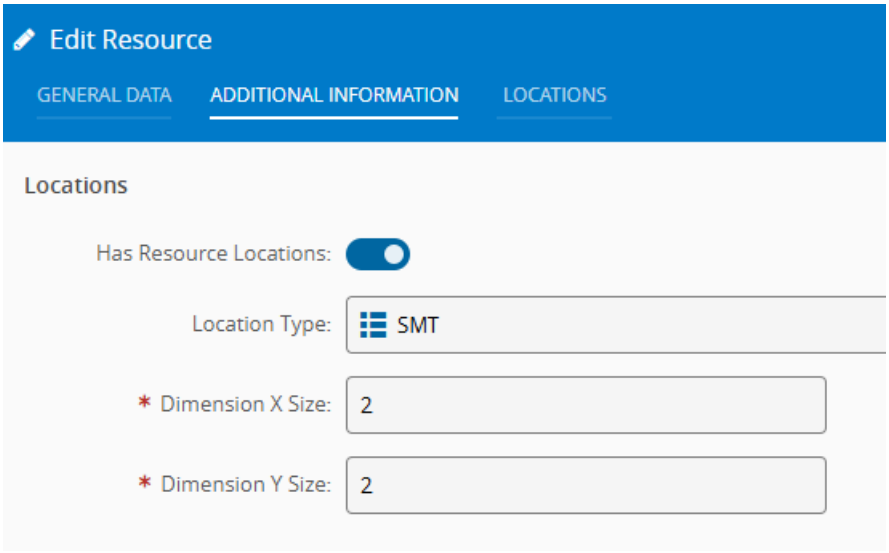
When docking a **Container of Resources** (for example feeder carts), the system automatically considers the Resources as consumable feeds in the docking Resource.

Example:

In SMT lines, component placement machines can handle multiple feeder carts.

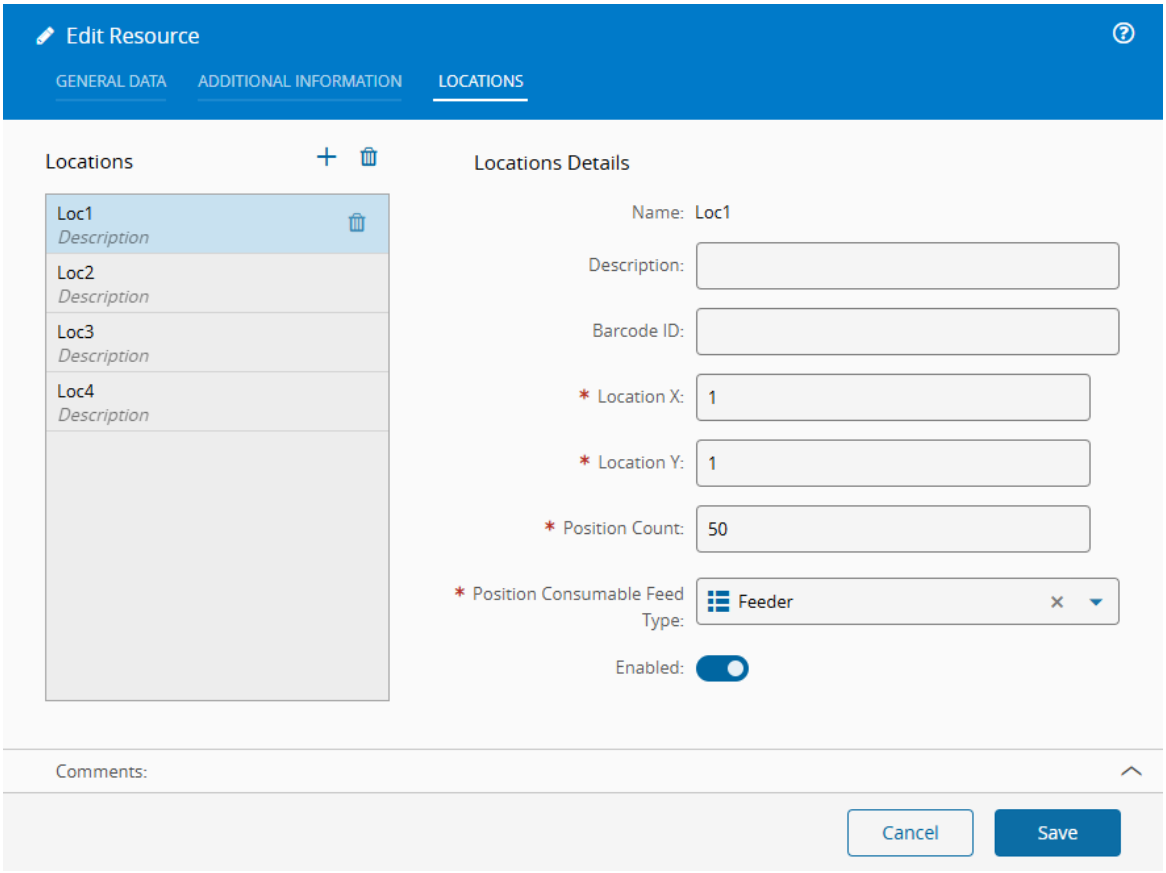
The docking Resource must have:

- **Has Resource Locations** enabled
- **Location Type** defined
- **Dimensions** configured



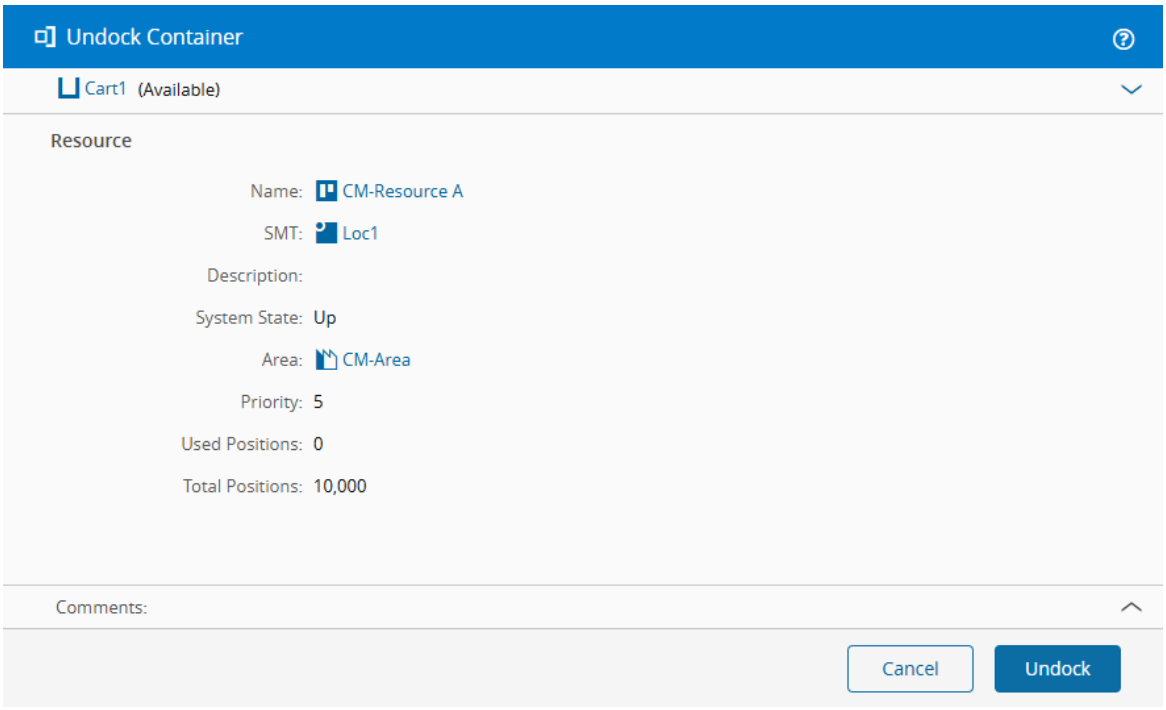
Locations must include:

- Location
- Position Count
- Position Consumable Feed Type
- Enabled status



Undock

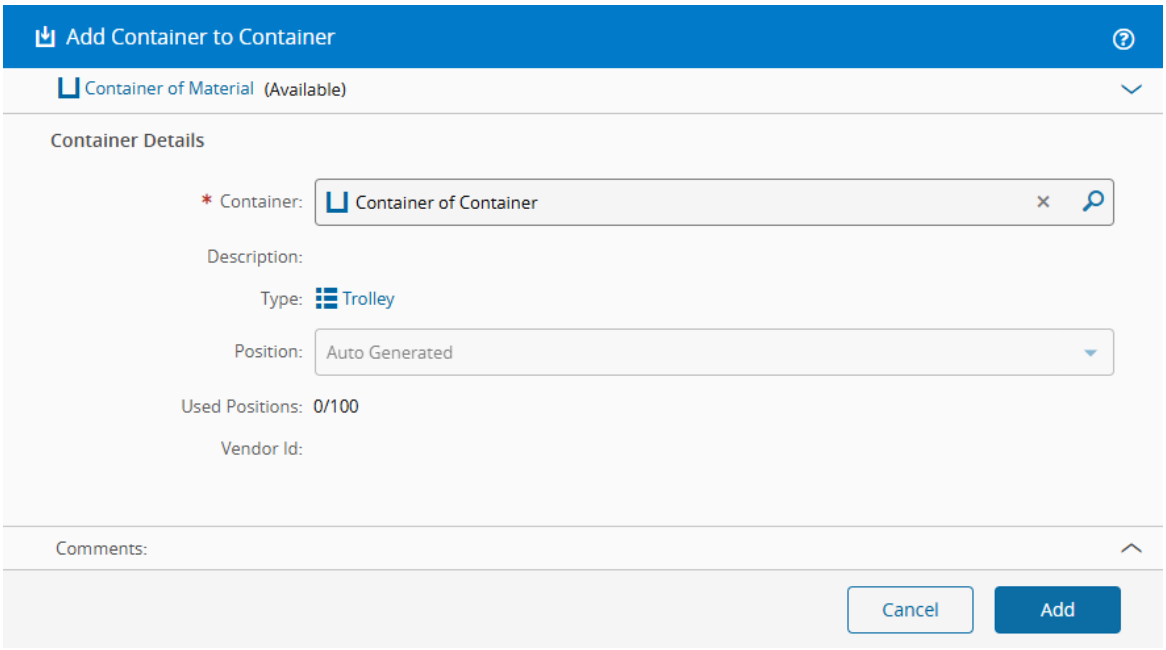
The **Undock** operation removes the Container from the Resource.



Add

Use the **Add** operation to assign the current Container to a **Parent Container**.

Define the relationship in the **Container Type Relation** Generic Table.



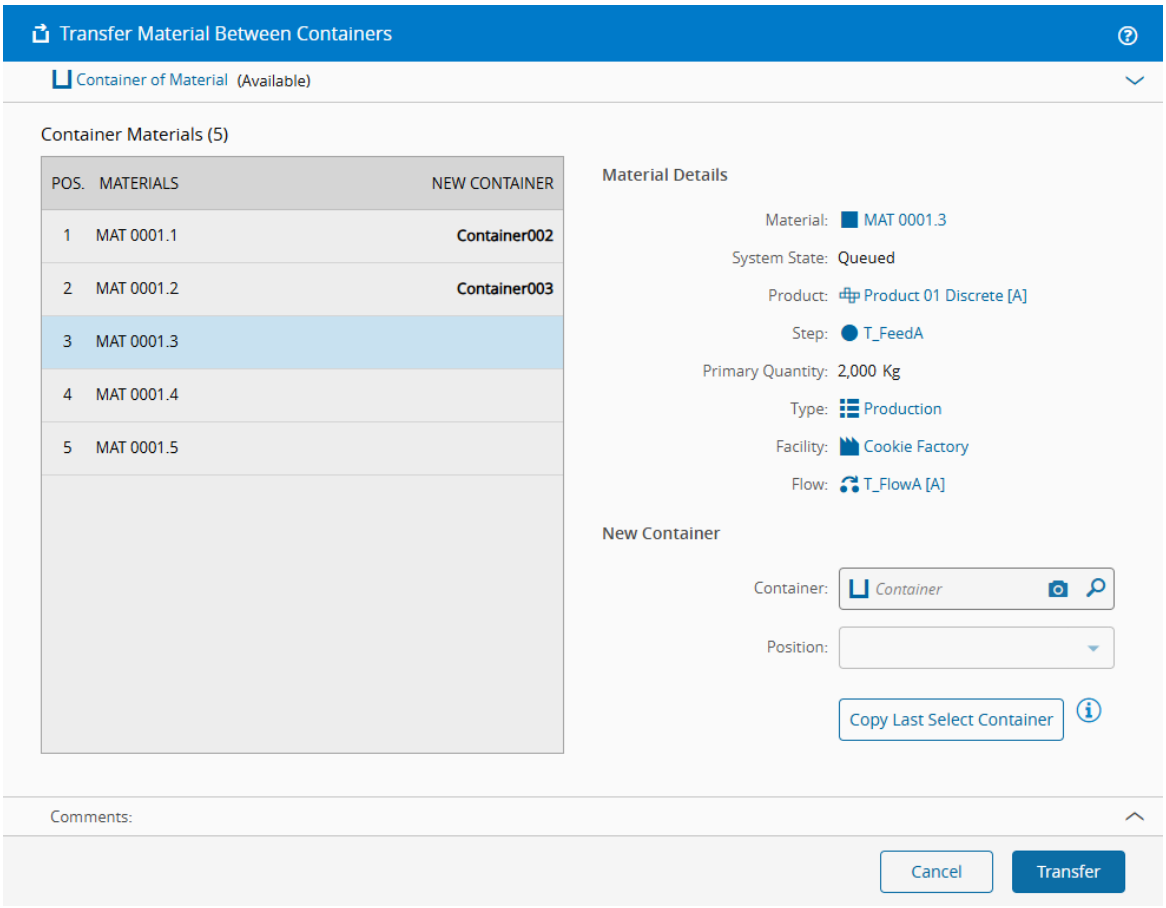
Transfer

The **Transfer** operation transfers the contents of a Container to another Container.

The operation name varies depending on the **Container Position Unit Type**:

- Transfer Materials
- Transfer Resources
- Transfer Containers

You can transfer each position to a different Container.



Make Unavailable

Sets the **System State** of the Container to **Unavailable**. In this state, the Container is not visible in selection lists.

Make Available

Changes the Container **System State** back to **Available**.

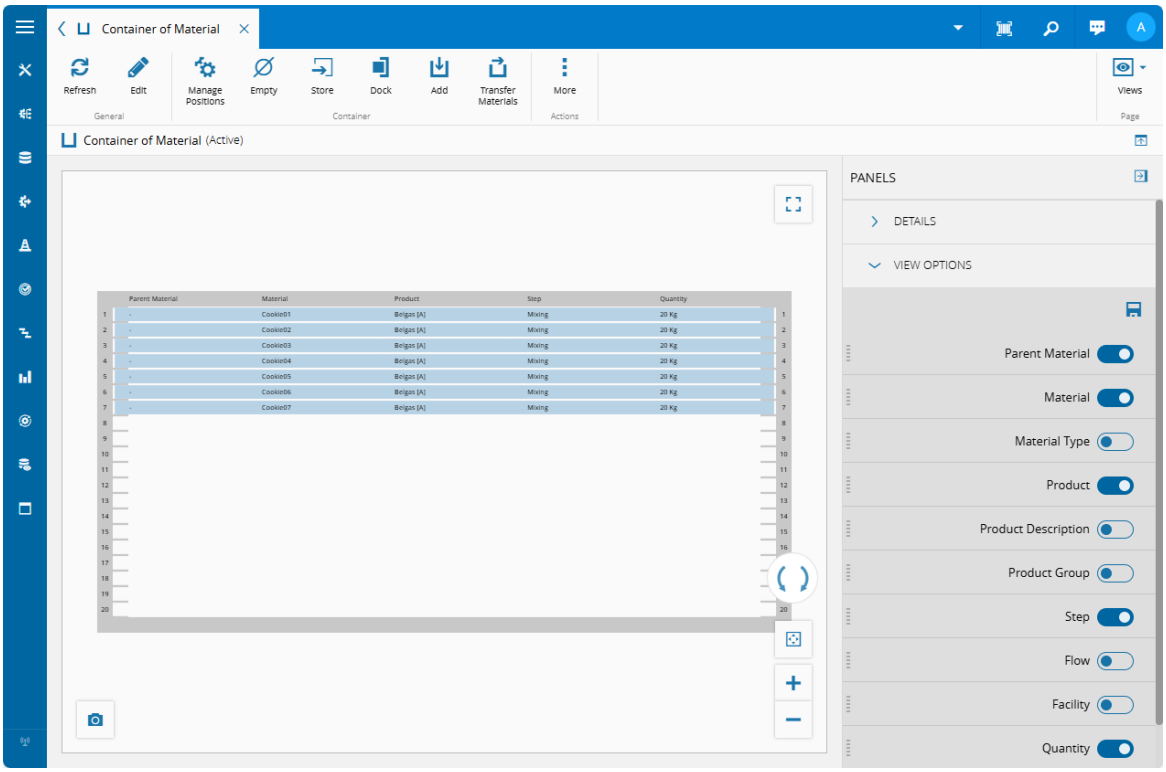
Container View

The **Container View** displays the Container layout and contents.

You can:

- Visualize positions and assigned objects
- View details of the selected position

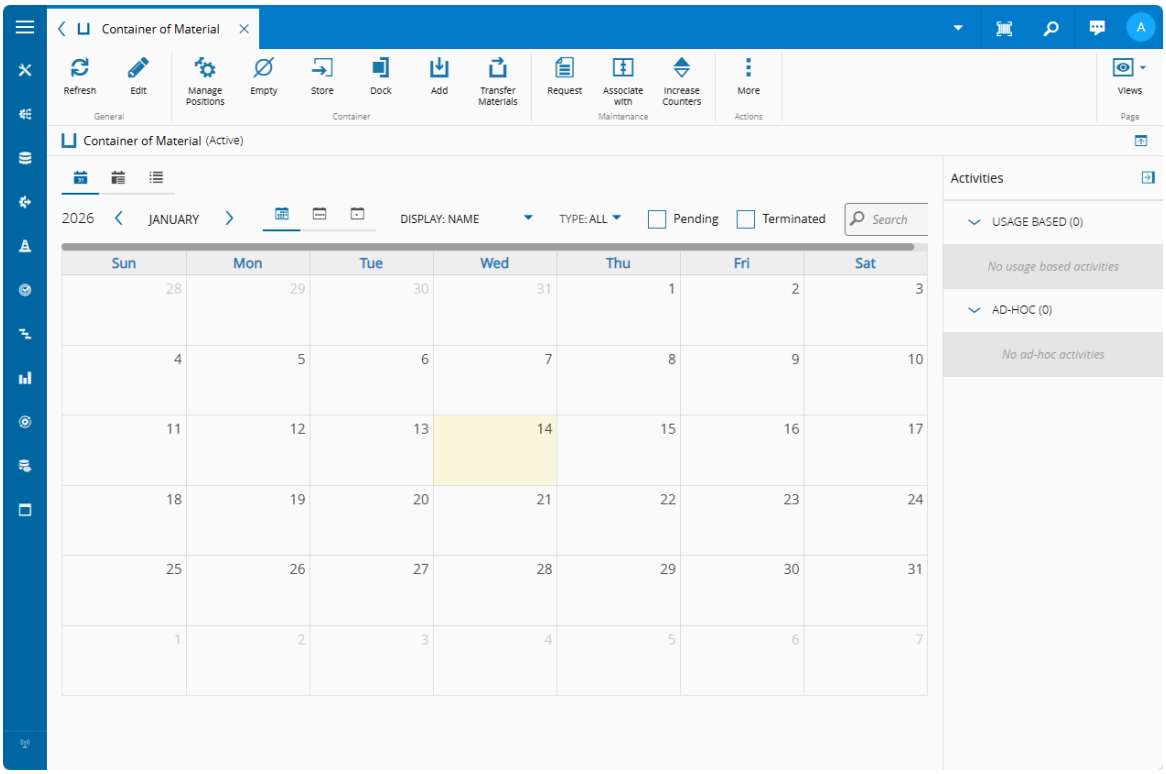
- Customize visible columns in **View Options**



Container Maintenance View

The **Container Maintenance View** displays scheduled and ad-hoc **Maintenance Action Orders (MAO)** related to the Container.

For more information, refer to Maintenance Management.



Material Container Type Context

The **Material Container Type Context** Smart Table defines restrictions for container usage.

You can restrict container types based on:

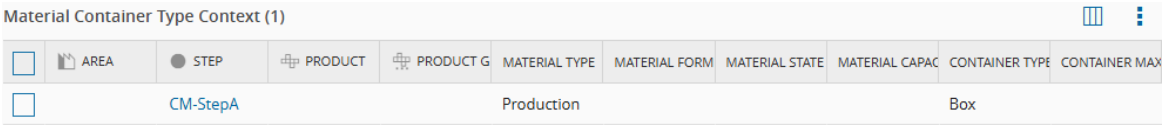
- **Area**
- **Step**
- **Product**
- **Product Group**
- **Material Type**
- **Material Form**
- **Material State**
- **Capacity Class**

You can also define the **maximum number of usable positions** for a Container type.

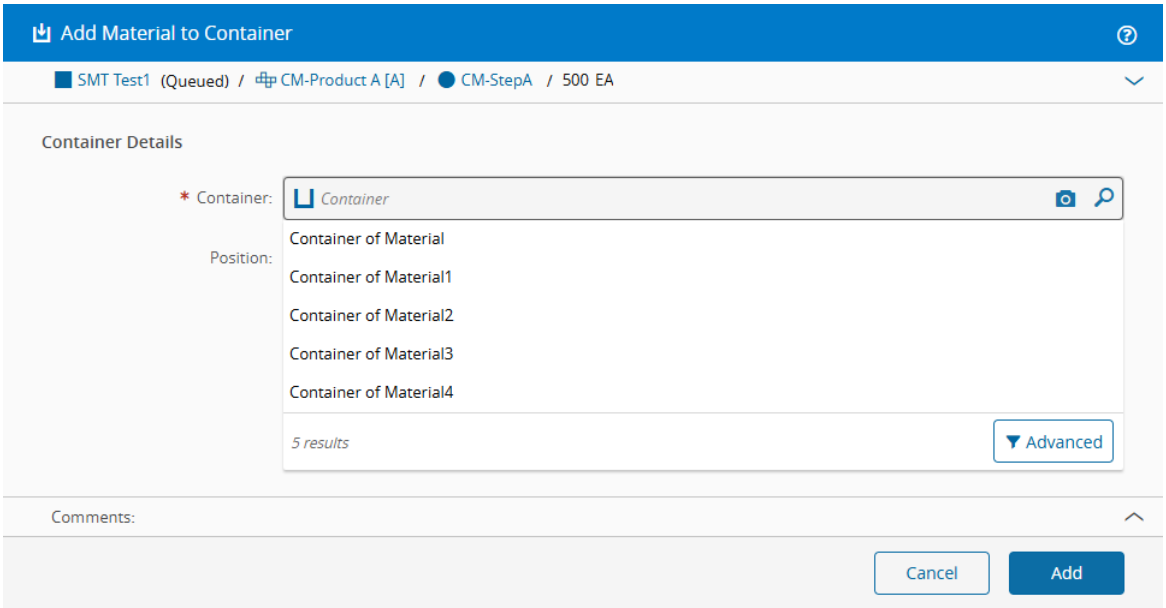
> **Note**
>
> To select a Step in this Smart Table, enable the Use Container Restrictions property.

Example

Example configuration using **Step**, **Material Type**, and **Container Type**.



In this configuration, when you attempt to add a **Material** of type **Production** at the selected Step, only Containers of type **Box** appear in the selection list.



Container

Name: Container of Material

Description:

Type: Box

Universal State: Active

System State: Available

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.