Critical manufacturing
an ASM PT company

Configuration

12.0.1

November 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.

Shift Planning Configuration

Copy Page

Estimated time to read: 13 minutes

Prepare the model introduced in Shift Planning before starting the scenarios. Follow the configuration steps in dependency order, or use the prepared master data.

Planner Role Features

An administrator assigned to `PlannerRole` can complete the entire tutorial. Before configuration, create this **Role** and provision **User** accounts for the planner and 16 operating **Employees**.

During configuration, use `PlannerRole` as the **Owner Role** of the **Certifications**, **Shift Plan**, and **Workgroups**. Link the planning **User** to `ControllerPlanner` and assign the operating **Employees** as its subordinates.

A separate planner account is optional. If it has no administrator access, assign it to `PlannerRole` and grant the features below; use an administrator for configuration.

Task	Features
Open **Business Data** and the **Shift Plan**	`BusinessData.NavigationPane`, `ShiftPlan.Show`
Inspect **Employees** and their **Certifications**	`Employee.Show`, `Certification.Show`
Manage **Employee** assignments	`ShiftPlan.ManageEmployees`
Generate **Employee Calendar Days**	`ShiftPlan.GenerateCalendarDays`
Manage absences, non-working time, pool placement, temporary assignments, and availability	`ShiftPlan.ManageCalendarDays`
Validate staffing	`ShiftPlan.Validate`

Task	Features
Inspect, edit, and restore the **Calendar Day** in the closure exercise	`Calendar.Show`, `CalendarDay.Update`
View the configured calendar-day types	`Administration.Tables`

Table: PlannerRole features for the planning exercises

Useful Documentation

Configuration

Follow the steps in dependency order to prepare the model for the exercises.

Organization and Calendar

Create the following objects from the **Business Data** menu, completing each row before continuing.

Step	Action	Value	Helpful Documentation
1	Create a **Calendar**.	`ProductionCalendar`. Do not add Shifts to the **Calendar** yet.	How to: Create a Calendar
2	Create a **Facility**.	`PortoFactory` must use `ProductionCalendar`.	How to: Create a Facility
3	Create an **Area**.	`ControllerProduction`, which belongs to `PortoFactory`.	How to: Create an Area
4	Create the **Resources**.	`ControllerAssemblyBench01`, `ControllerAssemblyBench02`, and `ControllerFunctionalTeste`	How to: Create a Resource

Step	Action	Value	Helpful Documentation
		`r01`, in the `ControllerProduction` **Area**.	
5	Create the **Steps**.	`AssembleController` and `TestController`, associated with the `ControllerProduction` **Area**.	How to: Create a Step
6	Create a **Flow**.	`ControllerFlow`, with **Flow** type **Sequential**. Add `AssembleController` followed by `TestController` **Steps**.	How to: Create a Flow

Table: Organization and Calendar foundation

Teams and Shifts

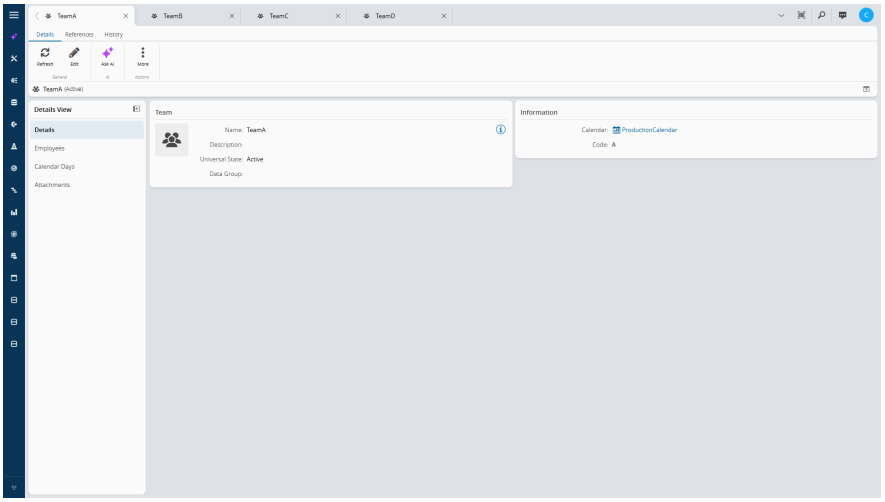
Before starting, confirm that `ProductionCalendar` exists, with the operational day starting at midnight on the current day.

1. In **Business Data** > **Team**, create the following **Teams**, selecting `ProductionCalendar` for each one. Select **Create** after entering the name, **Calendar**, and code for each **Team**.

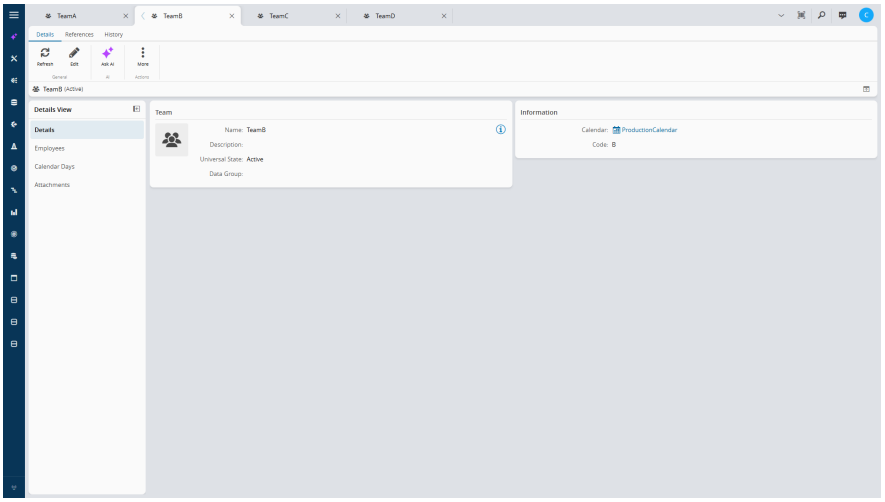
Team	Calendar	Code
`TeamA`	`ProductionCalendar`	`A`
`TeamB`	`ProductionCalendar`	`B`
`TeamC`	`ProductionCalendar`	`C`
`TeamD`	`ProductionCalendar`	`D`

Table: Teams and codes

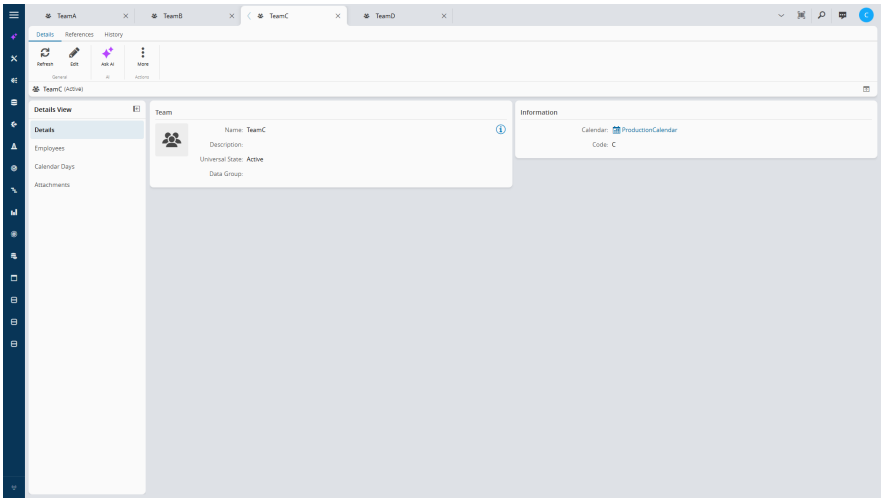
Team A



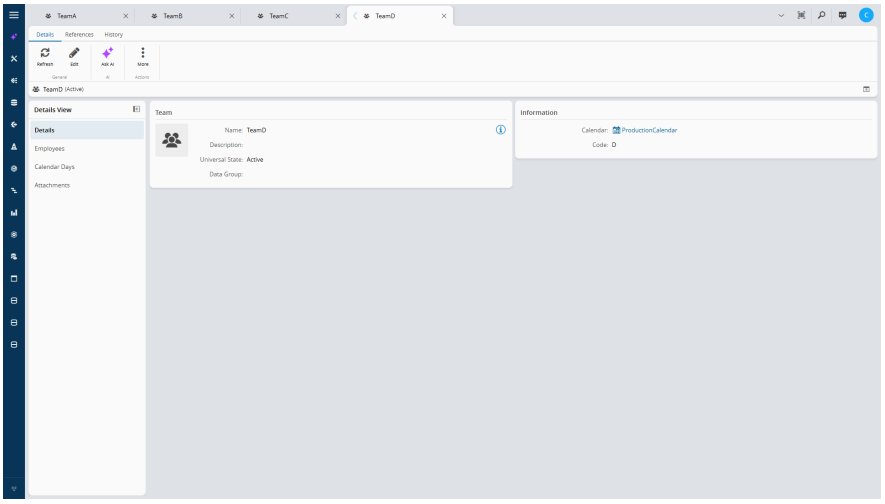
Team B



Team C



Team D



2. In **Business Data** > **Shift Definition**, create `ControllerDayShift` using `ProductionCalendar`. Add the three eight-hour Shifts below, keep **Cost Overtime Factor** at `1`, and leave **Non-working times** empty. You may customize the shift colors according to your preference.

Shift	Start Time	End Time	Code	Team Pattern
Shift 1	00:00	08:00	1	AADDCCBB

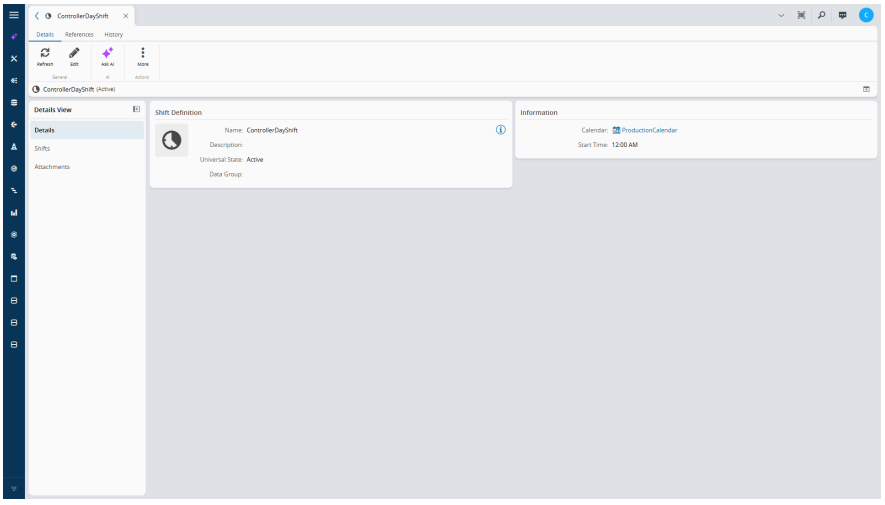
Shift	Start Time	End Time	Code	Team Pattern
Shift 2	08:00	16:00	2	BBAADDCC
Shift 3	16:00	00:00	3	CCBBAADD

Table: Shifts and TeamPattern values

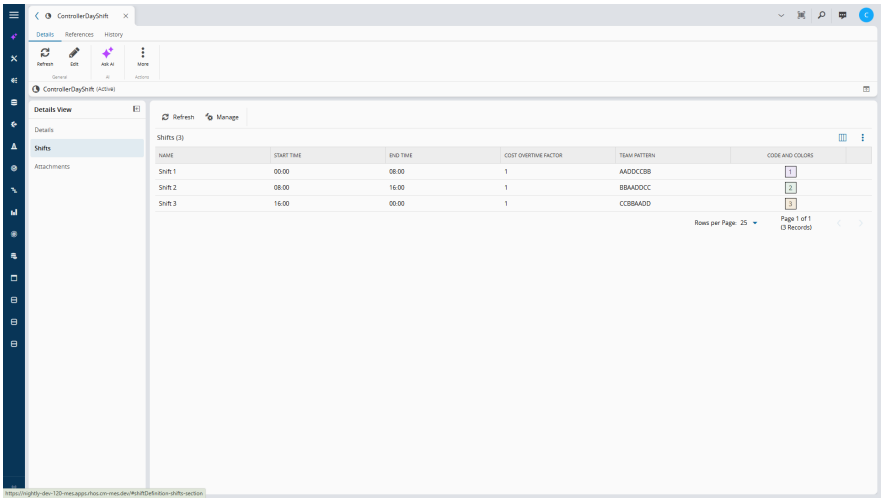
Each character identifies the **Team** for the corresponding day of the eight-day rotation. Reading the three patterns at the same position gives the **Teams** assigned to Shifts 1, 2, and 3 for that day; the remaining **Team** is off.

For example, position 3 gives D / A / B, with **Team** C off. All three patterns have eight codes, and all referenced **Teams** belong to the same **Calendar** as the **Shift Definition**.

Shift Plan Details

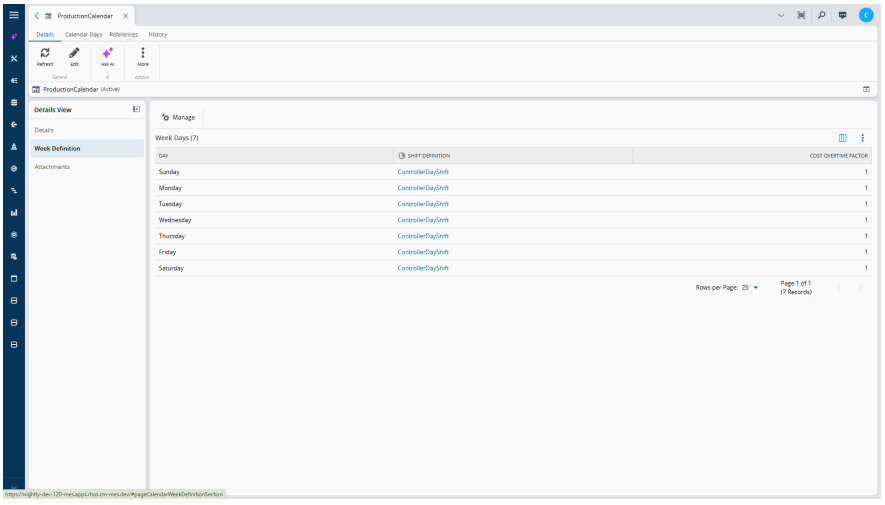


Shift Plan Shifts



3. Open `ProductionCalendar` > **Week Definition** > **Manage**, apply the `ControllerDayShift` to Monday through Sunday, and select **Save**.

   All seven weekdays use the complete **Shift Definition**. **Team** days off come from the rotation, rather than non-working weekdays in the **Calendar**.



Useful Documentation

Generate Calendar Days

Select a date interval for the exercise, allowing at least 16 consecutive days to inspect two complete eight-day cycles.

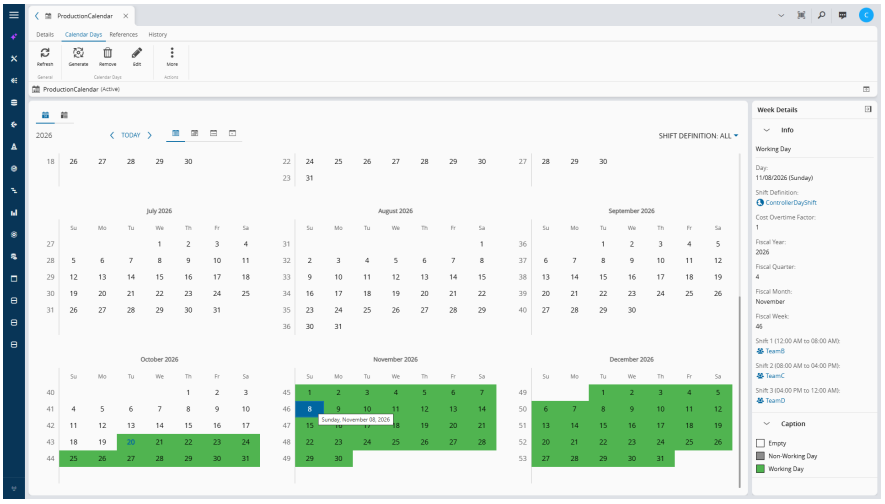
1. Open `ProductionCalendar` > **Calendar Days**, select the date interval, and select **Generate**.
2. Complete the wizard with the following values, and select **Generate**.

Setting	Value
Start Date	The first date of the selected interval.
End Date	The last date of the selected interval.
Number of days	Confirm the displayed count matches the selected interval.
Start Position	`1`, corresponding to the first character of each **Team Pattern**.
Existing **Calendar Days** handling	Generate only missing days; do not override existing days.
Fiscal information	Review the automatically calculated values against the fiscal settings of the **Calendar**.

Table: Calendar Day generation settings

Check the cycle boundary: day 8 has **Teams** B / C / D on Shifts 1 / 2 / 3 and **Team** A off; day 9 restarts with **Teams** A / B / C and **Team** D off. The same eight-day pattern repeats for longer intervals. Every date must have three staffed shifts, with no **Team** assigned to more than one shift that day.

Useful Documentation



Employees and Certifications

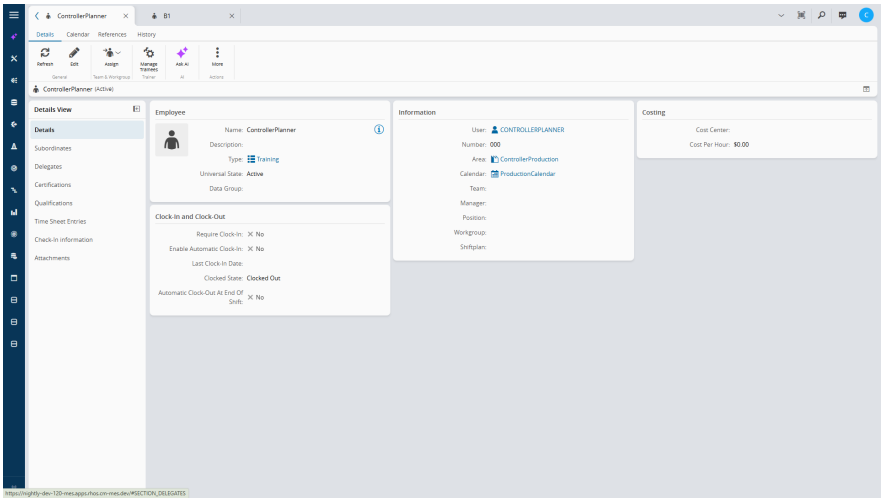
Create the planner and the 16 operating **Employee** records using the provisioned **User** accounts, then assign and grant the **Certifications** needed for their work. Use the actual account identifiers for the environment.

1. In **Business Data** > **Employee**, create `ControllerPlanner` first, then the operating **Employees** in the matrix below. Select the provisioned **User** for each **Employee** and apply these shared values:

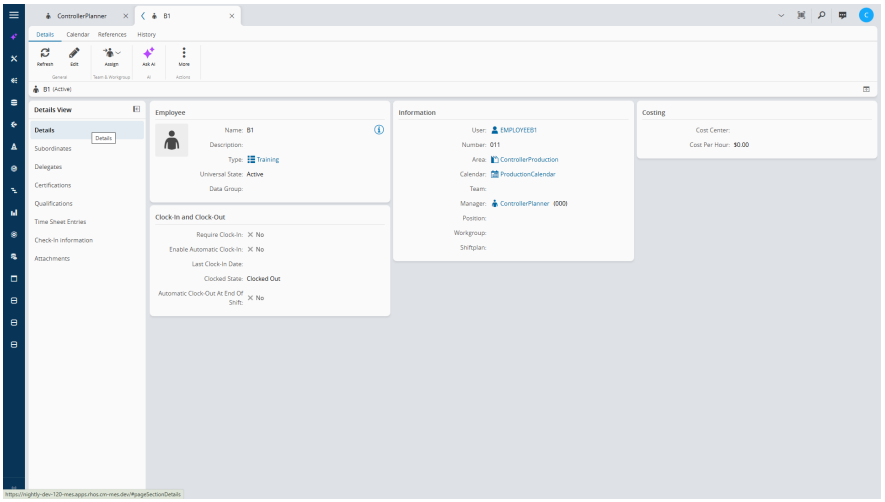
Setting	Value
Area	`ControllerProduction`.
Calendar	`ProductionCalendar`.
Employee Number	A unique identification number.
Type and **Cost Center**	The values approved for the environment.
Require Clock-In	`false` for this calendar exercise.

Table: Shared Employee settings

Shift Planner Employee



B1 Employee



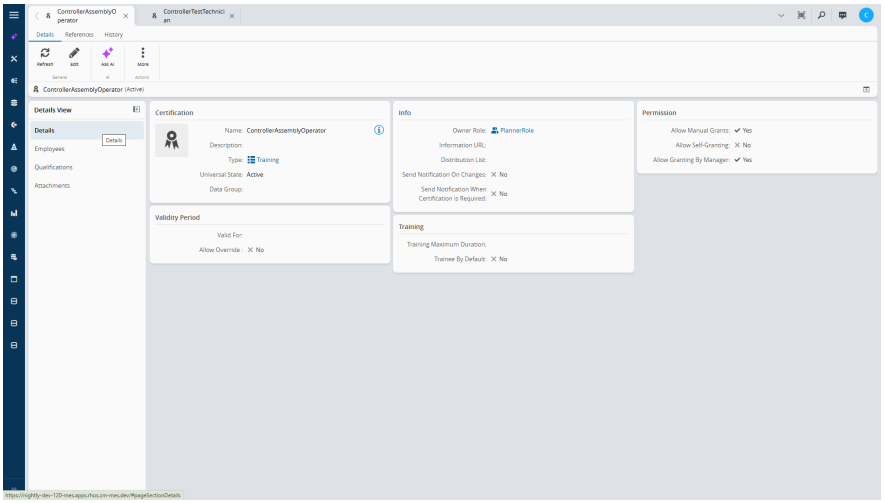
2. Open `ControllerPlanner` > **Subordinates** > **Assign**. Select all 16 operating **Employees** and select **Assign**. Verify that they are Active and that `ControllerPlanner` is their manager.
3. In **Business Data** > **Certification**, create `ControllerAssemblyOperator` and `ControllerTestTechnician`, applying the same settings to both:

Setting	Value
Type	The **Certification** Type approved for the

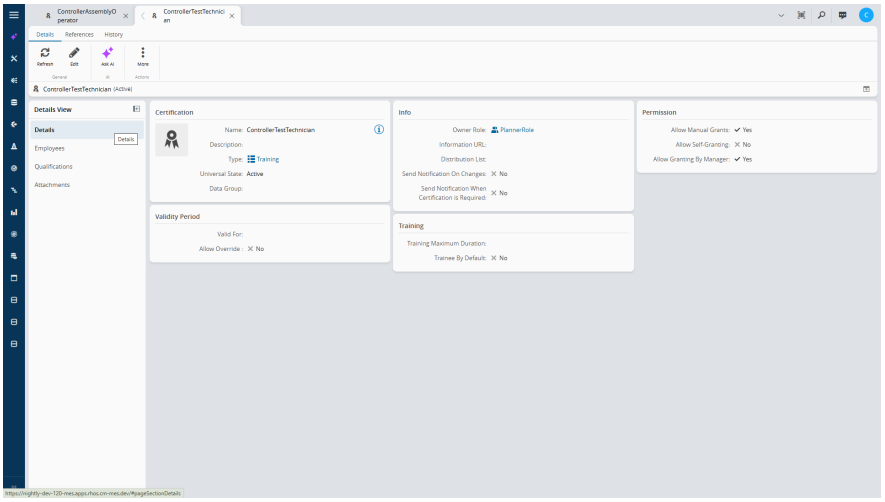
Setting	Value
	environment.
Owner Role	`PlannerRole`.
Allow Manual Grants	`true`.
Allow Self-Granting	`false`.
Allow Granting By Manager	`false`; this exercise uses owner-role authority.
Validity period	Unset.
Enable Trainees	`false`.

Table: Certification creation settings

Assembly Certification



Test Certification



4. Confirm that both **Certifications** are Active. On each **Certification**, open **Employees** > **Assign** and select the **Employees** in its column below. Leave **Granted** unset and select **Assign**. Verify the resulting **Assigned** state before granting.

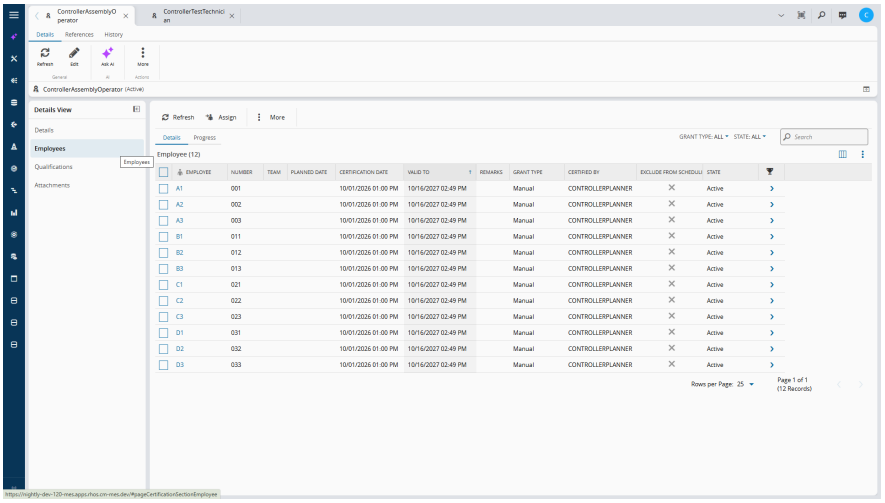
Team grouping	Assign ControllerAssemblyOperator to	Assign ControllerTestTechnician to
`TeamA`	`A1`, `A2`, `A3`	`A3`, `A4`
`TeamB`	`B1`, `B2`, `B3`	`B3`, `B4`
`TeamC`	`C1`, `C2`, `C3`	`C3`, `C4`
`TeamD`	`D1`, `D2`, `D3`	`D3`, `D4`

Table: Employee Certification assignment matrix

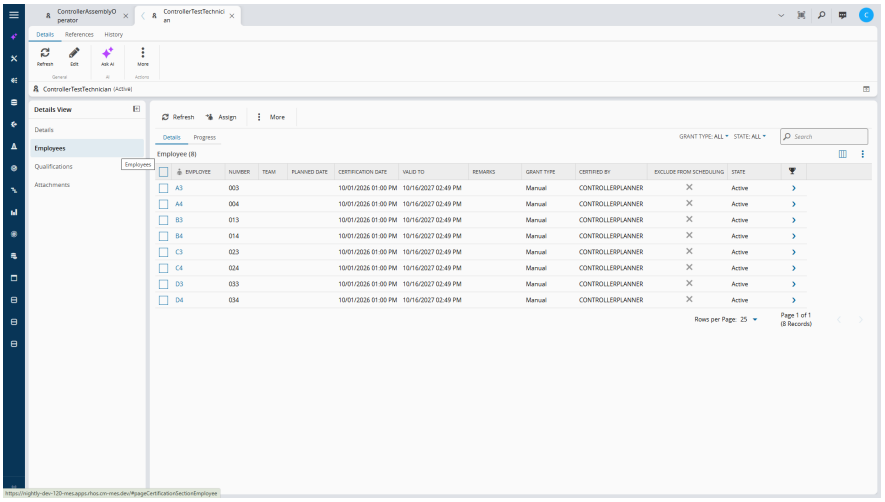
The **Team** column identifies the intended planning groups; this step assigns **Certifications**, not **Team** or **Workgroup** positions. The planner manages these **Employees** but does not receive the operating **Certifications** in this exercise.

5. In the **Employees** section of each **Certification**, select the assigned records and **Grant**. Verify that every association is **Active**. Assignment alone does not establish qualification; the grant gives the **Employee** the **Certification**.

Assembly Certification Employees



Test Certification Employees



Useful Documentation

Resource Personnel Requirements

Complete this optional step to calculate **Workgroup Personnel Requirements** from **Resource Personnel Requirements**.

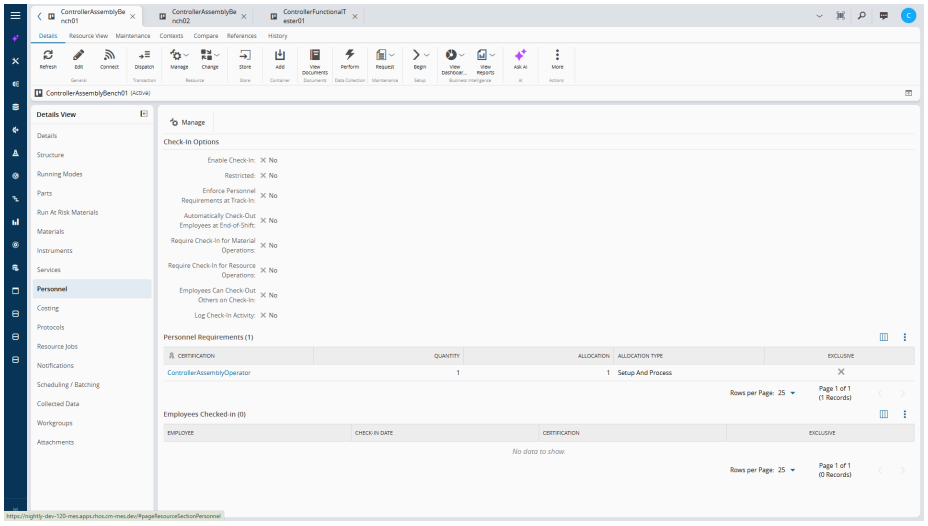
Open the **Personnel** section of each **Resource** and select **Manage**. Under **Personnel Requirements**, add the following **Certification** requirement to each **Resource**:

Resource	Certification	Quantity	Allocation	Allocation Type	Exclusive
`ControllerAssemblyBench01`	`ControllerAssemblyOperator`	`1`	`1`	`SetupAndProcess`	`false`
`ControllerAssemblyBench02`	`ControllerAssemblyOperator`	`1`	`1`	`SetupAndProcess`	`false`
`ControllerFunctionalTester01`	`ControllerTestTechnician`	`1`	`1`	`SetupAndProcess`	`false`

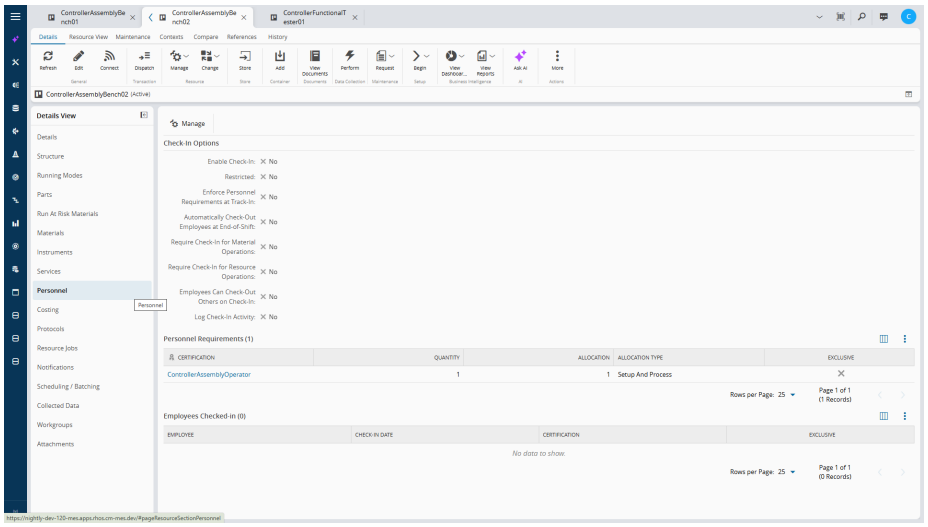
Table: Resource Personnel Requirements

Quantity `1` requires one qualified **Employee**; **Allocation** `1` means full allocation. Select **Update** on each **Resource** to save the settings and requirement.

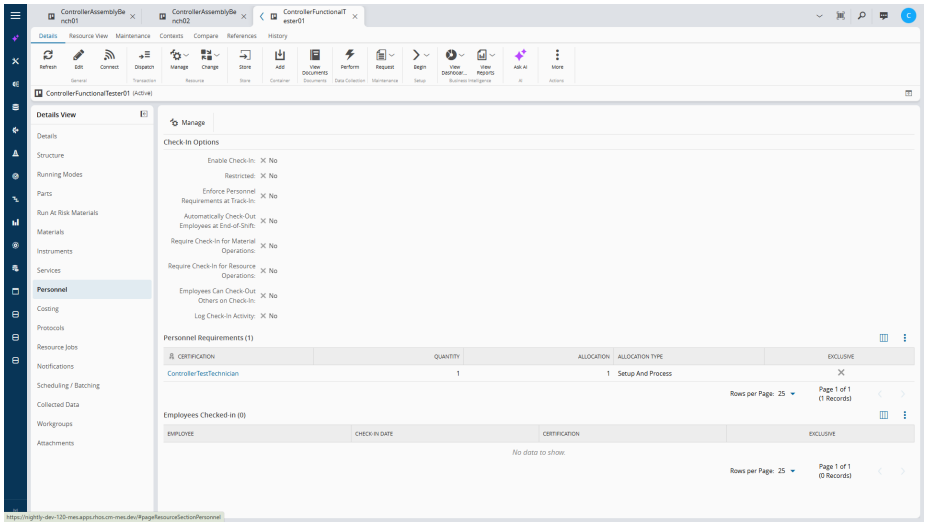
Assembly Bench 01



Assembly Bench 02



Tester



Useful Documentation

Shift Plan and Workgroups

1. In **Business Data** > **Shift Plan**, select **New** and complete the following settings:

Setting	Value
Name	`ControllerProductionShiftPlan`.
Calendar	`ProductionCalendar`.
Owner Role	`PlannerRole`.
Rules	Leave empty for this exercise.

Table: Shift Plan settings

2. Continue to **Workgroups**. Add `ControllerAssembly` and `ControllerFinalTest`, setting **Owner Role** to `PlannerRole` for both.
3. In **Workgroup Settings**, add the following **Resource** and **Step**/**Facility** associations:

Workgroup	Resources	Step	Facility
`ControllerAssembly`	`ControllerAssemblyBench01`, `ControllerAssemblyBench02`	`AssembleController`	`PortoFactory`
`ControllerFinalTest`	`ControllerFunctionalTester01`	`TestController`	`PortoFactory`

Table: Workgroup production associations

Add each **Resource** only once within the **Shift Plan**. Each **Step**/**Facility** pair is valid because its **Step** belongs to **Area** `ControllerProduction`, which belongs to `PortoFactory`; keep each pair unique within its **Workgroup**.

This tutorial does not cover the use of rules in **Shift Plan** requirements. Any rules added when configuring the **Shift Plan** must use the Shift Planning scope to validate working time requirements.

> **Note**
>
> Configure the workgroup within the Shift Plan entity. Do not configure the Workgroup entity itself for this step.

4. For each **Workgroup**, enter the following **Personnel Requirements** explicitly:

Workgroup	Position (Certification)	Quantity
`ControllerAssembly`	`ControllerAssemblyOperator`	`2`
`ControllerFinalTest`	`ControllerTestTechnician`	`1`

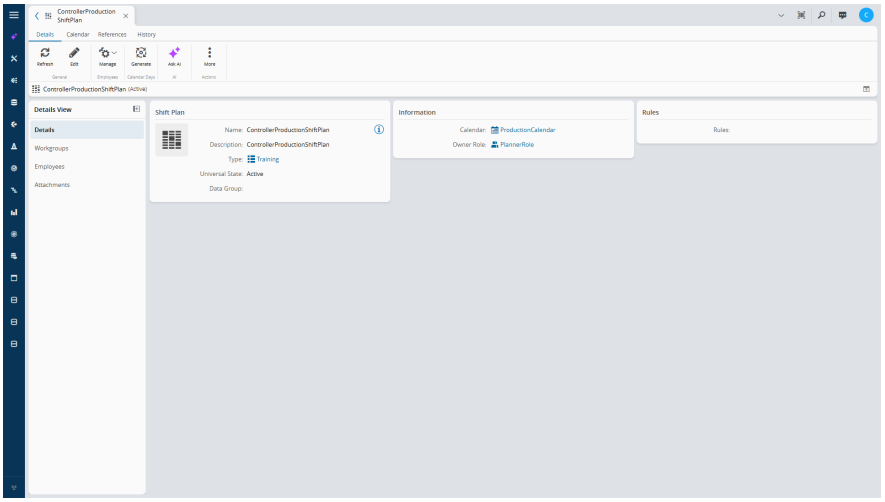
Table: Workgroup Personnel Requirements

Resource requirements describe individual equipment; **Workgroup** requirements define staffing for the group. They are configured separately.

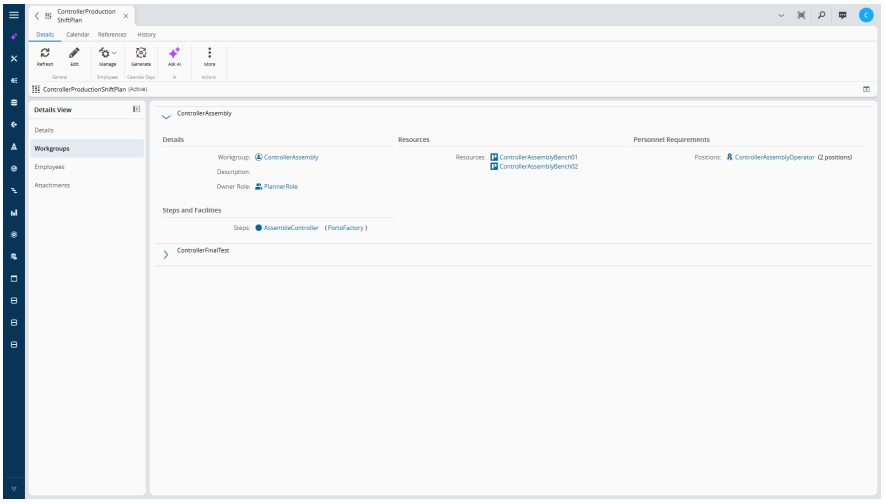
> **Note**
>
> If you completed the preceding Resource Personnel Requirements step, select Calculate Personnel Requirements for each Workgroup instead of entering the requirements manually. Confirm that the calculated quantities match the table above.

5. Select **Create** and reopen `ControllerProductionShiftPlan` to check the saved configuration.

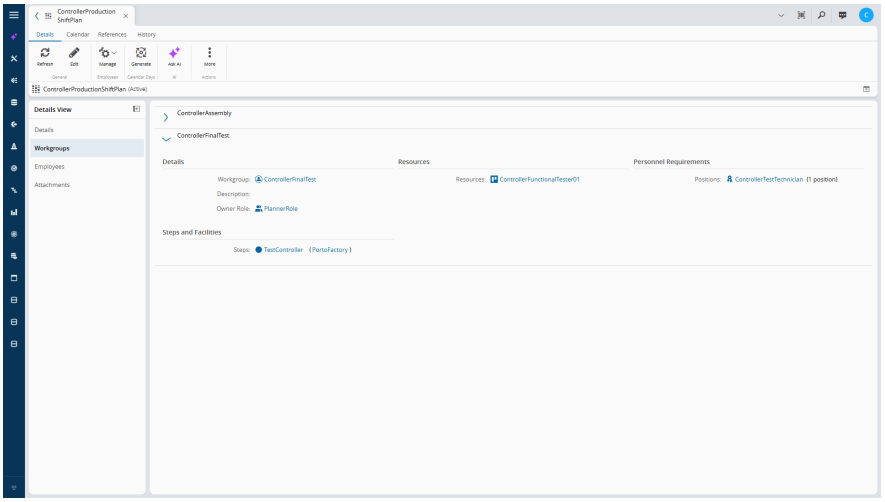
Shift Plan Details



Shift Plan Assembly Workgroup



Shift Plan Test Workgroup



Useful Documentation

Shift Plan Calendar Days

Before starting, confirm that the **Shift Plan** and all operating **Employees** are Active, their required **Certifications** are Active, and the selected interval already has **Calendar Days**. Continue as the **User** linked to `ControllerPlanner`. If you prepared the model with a separate administrator account, switch to that planner **User** now.

1. Open `ControllerProductionShiftPlan` and select **Manage Employee Teams and Workgroups**.

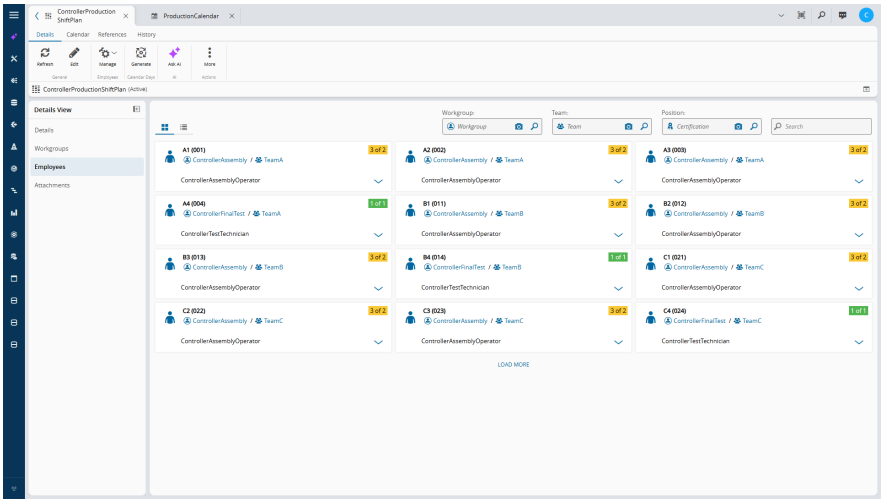
Employees within each Team	Workgroup	Position
Employees 1, 2, and 3	`ControllerAssembly`	`ControllerAssemblyOperator`
Employee 4	`ControllerFinalTest`	`ControllerTestTechnician`

Table: Normal assignments for every Team

Set the **Team** of each **Employee** to the corresponding `TeamA`, `TeamB`, `TeamC`, or `TeamD` from the matrix. Set **Workgroup** and Position together, using a

Certification the **Employee** currently holds. Keep each designated replacement in Assembly for the baseline.

2. Inspect **Summary** for all four **Teams**. Each must have three Assembly Operators against a requirement of two, and one Test Technician against a requirement of one. Select **Save and Close**.



3. Select **Generate Shift Plan Calendar Days** and use these settings:

Setting	Value
Start Date and **End Date**	The same interval used for **Calendar Day** generation.
Override employee non-working days	Disabled.
Override employee non-working times	Disabled.
Override employee temporary assignments	Disabled.

Table: Employee Calendar Day generation settings

Select **Generate**.

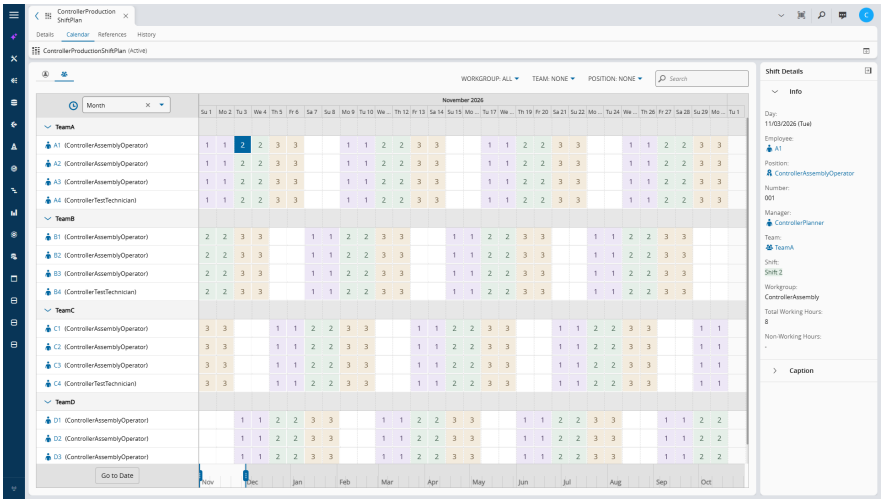
Earlier **Calendar Day** generation defined the dated shifts and **Team** rotation; this separate operation creates individual **Employee Calendar Days** from those records and the saved **Team**, **Workgroup**, and Position assignments.

4. Open the **Calendar** view of the **Shift Plan** and select both **Workgroups** in the filters. Select the third date of the generated interval as the exercise date, counting the start of the interval as day 1. With **Start Position** set to `1` during **Calendar Day** generation, verify the **Employee Calendar Days** against the following table:

Shift / status	Team	Employees	Expected coverage
Shift 1: midnight-8:00 AM	`TeamD`	`D1` - `D4`	Three Assembly Operators and one Test Technician.
Shift 2: 8:00 AM-4:00 PM	`TeamA`	`A1` - `A4`	Three Assembly Operators and one Test Technician; use this **Team** and shift for the exercises.
Shift 3: 4:00 PM-midnight	`TeamB`	`B1` - `B4`	Three Assembly Operators and one Test Technician.
Off-duty for the operational day	`TeamC`	`C1` - `C4`	**Employee Calendar Days** exist, but the **Employees** are not scheduled to work.

Table: Exercise-date verification

Confirm that all 16 **Employees** have records and that no absence, pool, availability, or temporary-assignment exceptions affect the exercise baseline. Each operating shift deliberately has one more Assembly Operator than required.



Useful Documentation

Non-Working Calendar Day and Time Type

As the administrator, prepare the following non-working day and time types before starting the scenarios. These custom entries are separate from the nine system calendar-day types. If you use a separate planner account, return to it for the scenarios.

1. Open **Administration** > **Tables** > **Lookup Tables** and select `EmployeeCalendarNonWorkingTimeType`.
2. Open the **Data** tab and check whether `Absence-Hours` already exists. If it does not, select **Add** and enter the following values. Preserve the existing system entries.

Field	Value
Name	`Absence-Hours`
Description	`Absence-Hours`
Enabled	`Yes`

Table: Non-working time lookup entry

3. Open **Administration** > **Tables** > **Lookup Tables** and select `CalendarDayType`.

4. Open the **Data** tab and check whether `Absence` already exists. If it does not, select **Add** and enter the following values. Preserve the existing system entries.

Field	Value
Name	`Absence`
Description	`Absence`
Enabled	`Yes`

Table: Non-working day lookup entry

5. Open **Administration** > **Tables** > **Generic Tables** and select `CalendarDayType`.
6. Open the **Data** tab and check whether `Absence` already exists. If it does not, select **Add** and enter the following values. Preserve the existing system entries.

Field	Value
Type	`Absence`
IsWorkingDay	`false`
Code	`A`
BackgroundColor	Choose a color for this absence type.
ForegroundColor	Choose a text color that contrasts with the background.
IsSystem	`false`

Table: Custom non-working calendar-day type

7. Select **Save**. If the entry already existed, verify its non-working and non-system settings instead of adding a duplicate.

Useful Documentation

Master Data

The tutorial master data was prepared and tested in Critical Manufacturing MES 12.0 and provides the configured model described in this tutorial.

If you use the master data instead of completing the configuration steps, select a date interval and run **Generate Calendar Days** on `ProductionCalendar`. Then run **Generate Shift Plan Calendar Days** for the same interval.

This is the Master Data file used to create this model.

Next Steps

With the model configured, continue with these scenarios in order:

1. Calendar Day Types - inspect the calendar day types.
2. Cover Employee Absence - cover an absence with a qualified temporary assignment and verify staffing.

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.

Copyright © 2023 Critical Manufacturing. All rights reserved.