Critical manufacturing
an ASM PT company

Scoping

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.

Scoping an Integration

Copy Page

Each equipment integration is a separate problem, and even equipment from the same supplier can vary. See Introduction. The effort of an integration still comes from a known set of drivers. Use this page to explain to the customer where that effort goes, and to collect the answers that decide it before you estimate. The page gives no effort figures: estimate from the answers and from the history of your own projects.

Effort Drivers

Effort driver	What raises the effort	More information
Protocol and driver	A driver that ships no driver definition, such as SECS/GEM, OPC UA, MQTT, TCP/IP, or REST Client, needs every ID collected from the equipment and declared. Equipment that fits no shipped driver needs a custom driver. The driver is chosen once: the driver definition, the parameters, and every equipment task are written for one protocol.	Communication Drivers
Characterization and machine time	The documentation of the equipment names its messages, but not always when the equipment produces them. The real behavior must be checked on the equipment, and machine time for tests is usually limited.	Basic Concepts
Use cases and MES interoperability	Each use case is a workflow of its own. Each use case that calls MES adds services, replies, and timeouts to design and test.	Connect IoT - MES Interoperability Scenarios Tutorial

Effort driver	What raises the effort	More information
Variation per machine	Machines that differ only in their settings share one controller, with the settings kept in MES. Machines that differ in behavior need logic of their own.	Integration Landscape Design
Custom code	A Code task, a custom task, a converter, or a custom driver must be written, tested, versioned, and released as a package.	Developer Portal ↗
Simulators for testing	Without a simulator, every test needs the equipment. The Test Orchestrator emulates MES and provides simulators for many shipped protocols.	Test Orchestrator simulators ↗
Cost per release	An upgrade of MES can bring steps of its own for each integration, such as moving tasks to **Automation Tasks Libraries**, migrating deprecated tasks, and rolling a new controller version out to every instance.	Connect IoT 12.0 Upgrade Checklist ↗

Table: Effort drivers of an equipment integration

No fixed effort ranges apply per driver or per use case, because the effort depends on how the integration is used. Estimate it from the effort drivers above, and plan tests in conditions as similar as possible to production, to confirm that the integration meets the non-functional requirements.

Characterization Checklist

Go through these questions with the equipment owner and the process owner. An open answer is a risk to name in the estimate.

Equipment and Connection

- How does the equipment offer its data: files in a folder, a GEM host port, an OPC UA server, an MQTT broker, an HTTP API, or text on a socket?

- Who starts the conversation: the driver, or the equipment? For a GEM tool, which HSMS role does its manual state?
- Does another host, such as a vendor tool, already use the one host connection that a GEM tool accepts?
- Which network path and ports are open between the equipment and the **Automation Manager**, and in which direction?
- Does the driver need a Windows host, a device attached to the computer, or a mounted folder?

Data, Events, and Commands

- For a driver that ships no definition, does the equipment document every ID: variables, collection events, reports, and remote commands for SECS/GEM, NodeIds for OPC UA, topics and payload encoding for MQTT, the frame format for TCP/IP, or the endpoints of a REST API?
- When does the equipment raise each event in the real process, and what happens to a file after it is written: is it appended to, moved, or deleted?
- Which commands does each use case send, in which equipment state, and what does the equipment reply?
- Which values does MES need, at which moment, and in which unit and format?

Use Cases and MES

- Which use cases does the integration cover, for example data collection, tracking, recipe selection, or interlocking, and which MES services does each one call?
- Which use cases stop production when MES does not answer, and how long can each one wait?

Variation and Scale

- How many machines are in scope, and do they share the model and the software version?
- What differs between the machines: only settings, such as a folder or an address, or also behavior?
- Does any use case need logic that no task of the product provides?

Testing and Operation

- How much machine time is available for characterization and tests, and is there a simulator for the protocol?

- Which maintenance windows exist, and which changes must roll out without stopping production? See What Is Automation Scheduled Action?.
- Who operates the integration after go-live, and how often is MES upgraded? See Connect IoT Runbook.

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.