Critical manufacturing
an ASM PT company

Real-Time Insights on Shop-floor Hierarchy

11.2

February 2026

DOCUMENT ACCESS

Public

Real-Time Insights on Shop-floor Hierarchy

Estimated time to read: 10 minutes

Overview

Manufacturing experts understand processes, while technical experts understand data structures. Bridging this gap is crucial to turning raw data into actionable insights. To achieve this, Critical Manufacturing developed a Canonical Data Model (CDM) anchored in the ISA-95 standard. The CDM acts as a universal translator, aligning complex technical systems with real business needs.

ISA-95 provides the common terminology and hierarchical structure - enterprise, site, area, work center, and work unit - that manufacturing professionals already recognize. Building on this, the CDM shifts from a static, table-based approach to an event-driven model. Instead of fragmented technical entities, data is expressed as meaningful events - covering materials, resources, production, maintenance, and calendars - each event self-contained and carrying its full context.

This design delivers two key outcomes:

- **Real-time visibility** - operational dashboards update instantly as events occur, enabling immediate insights into production performance.
- **Cross-plant comparability** - standardized event structures allow benchmarking across facilities with different underlying systems.

The CDM also integrates with Unified Namespace (UNS) architectures. UNS replaces fragile point-to-point integrations with a centralized, hierarchical information model that becomes the single source of truth. Built on a Publish/Subscribe messaging pattern - commonly implemented with MQTT (Message Queuing Telemetry Transport) - UNS enables systems to publish data to standardized topics and consume only what they need, creating a flexible, event-driven communication layer.

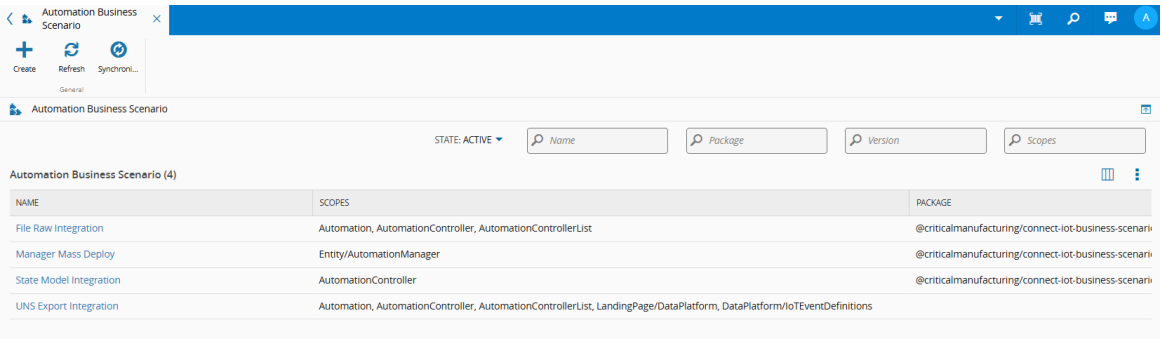
In this tutorial, we will show you how to configure your MES to use CDM events and UNS, providing real-time insights into your shop-floor operations.

Preconditions

When installing the MES environment:

- The IoT Data Platform Advanced module must be installed and available.
- The MES Analytics feature must be enabled.
- Light CDM Events must be enabled.
- The UNS must be enabled and the MQTT broker must be properly configured.
- The UNS Export Integration business scenario must be configured.

> **Note**
>
> The UNS Export Integration business scenario is included with the installation when the MES Analytics feature is enabled. However, it becomes functional only if Light CDM Events are enabled and the UNS with its MQTT broker is properly configured.



Steps to Implement Real-Time Insights

Enable Light CDM Events and UNS

CDM Events are the foundation of the Critical Manufacturing MES data strategy. Instead of relying on static database tables, CDM represents manufacturing activity as standardized, event-driven messages.

These events are continuously generated and distributed:

- The HouseKeeper service produces CDM events based on replication events and publishes them to Kafka.
- CDM events serve as the single source of data for reports and dashboards created with Grafana, Stimulsoft Reports, and CubeJS.
- When configured, CDM events can trigger automated workflows in the Data Platform.

Light CDM Events are a simplified variant, designed for real-time performance by including only the essential event properties. They provide a balance between efficiency and completeness, making them particularly well-suited for UNS integration.

To enable Light CDM Events and UNS:

1. Follow the steps in the Installation guide.
2. In the same section, enable the UNS container deployment.
3. Configure the MQTT broker connection.

Adding Additional Properties to Light Events

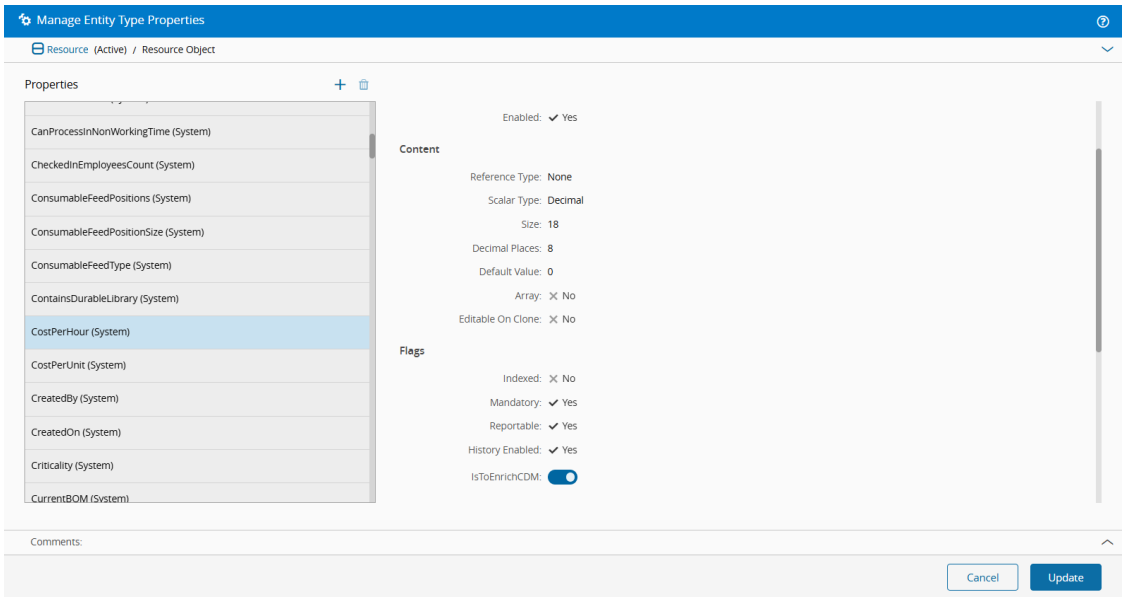
Critical Manufacturing MES ships with several CDM Events. Among them, only three have the **Has Light Variant** property enabled:

- Material Operations
- Resource State Change
- Post Data Collection

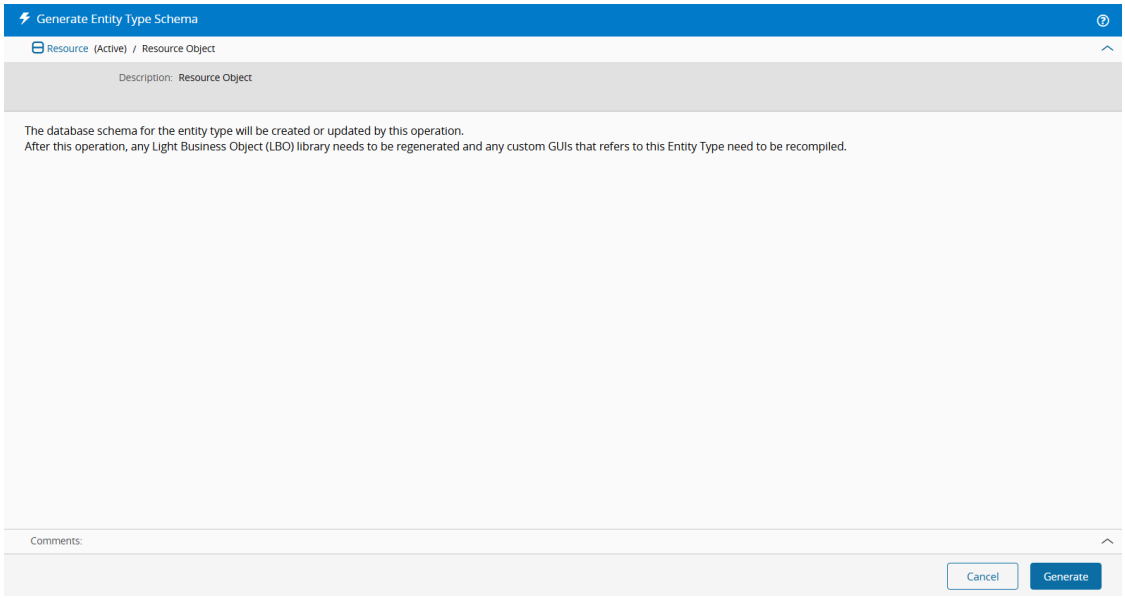
For these events, the properties included in the light variant are documented in the MES Canonical Event Definition. If you need additional data beyond the default set, you can enrich light events by enabling extra properties.

To add an additional property to a CDM event, you should:

1. Navigate to **Administration** > **Entity Type**.
2. Select the relevant Entity Type (for example, `Resource`).
3. Open the **Properties** tab and choose the **Manage** button.
4. Locate the desired property (for example, `CostPerHour`) and enable the **Is To Enrich CDM** toggle.



5. Select **Update**.
6. Select the **Generate** on the top ribbon, then select Generate again on the wizard that appears so the system will regenerate the schema so that the property is included in event generation.

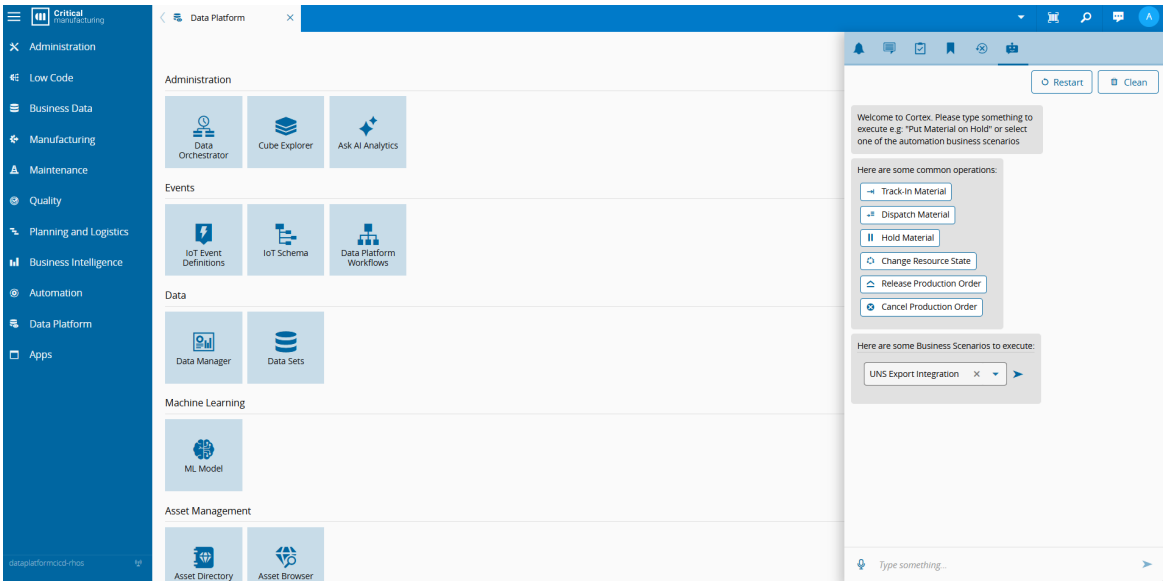


Using Cortex to Create Automation Controller

When creating an Automation Business Scenario, the first step is to select a scope. The scope determines which pages the Cortex chatbot can access for that business scenario.

For the `UNS Export Integration` business scenario, one of the available scopes is `LandingPage/DataPlatform`. To proceed:

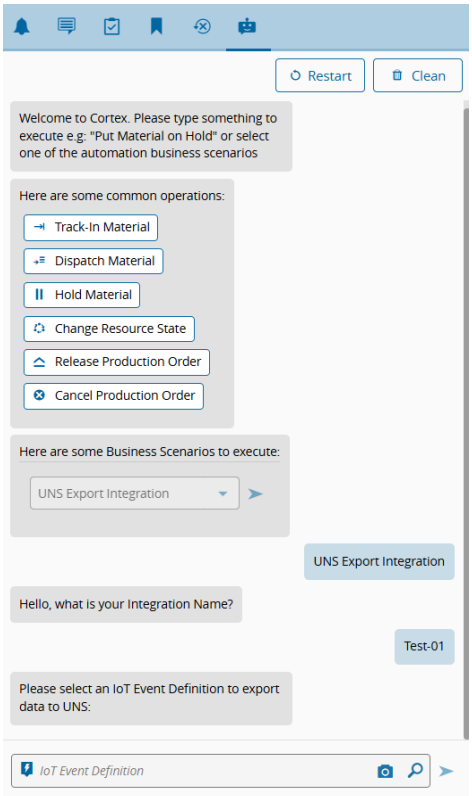
1. Navigate to the **Data Platform** menu entry.
2. Open the **Activity Center** and select the **Chatbot** tab.



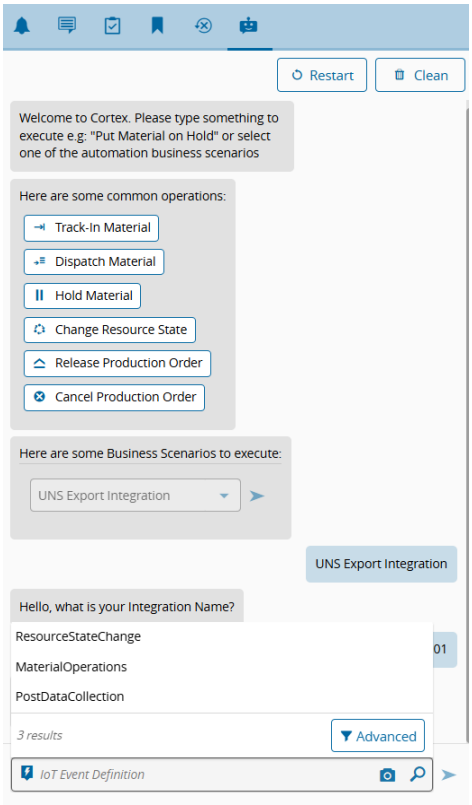
In the chatbot, you can access which business scenarios are available for the current page. In this case, select the `UNS Export Integration` business scenario and confirm your choice with the ➤ icon.

> **Note**
>
> If the chatbot was already open in another scope, you may need to select Restart after switching to a Data Platform scope to ensure the `UNS Export Integration` business scenario is loaded correctly.

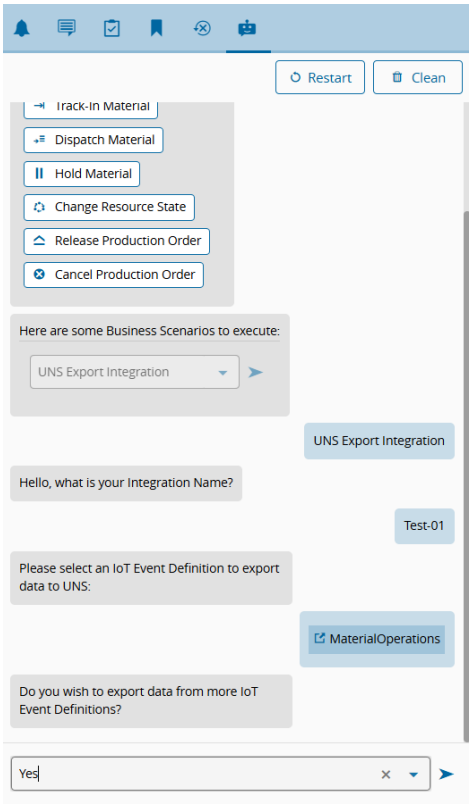
Next, the system prompts you to provide a name for the new Automation Controller. Choose a clear, descriptive name that reflects its purpose. For this tutorial, we will use `Test-01`.



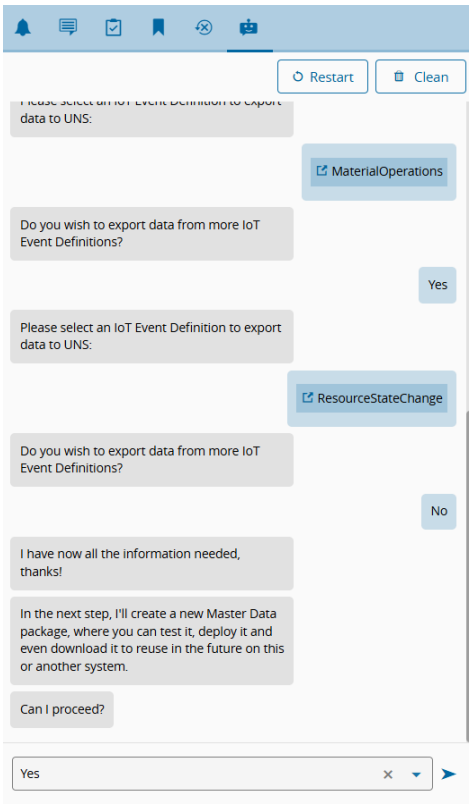
The system then asks you to select an IoT Event Definition. Only definitions with Has Light Variant enabled are displayed, as shown in the image below.



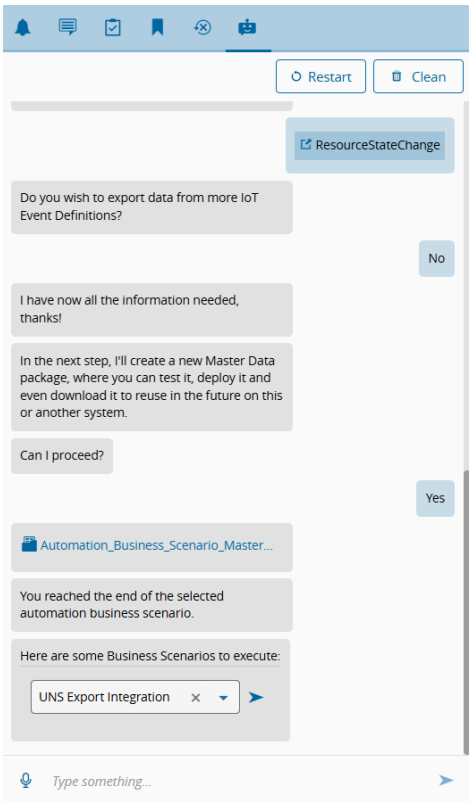
You may select more than one IoT Event Definition, and the system will guide you through this selection. In this tutorial, we chose the `ResourceStateChange` and the `MaterialOperations`.



Selecting ResourceStateChange



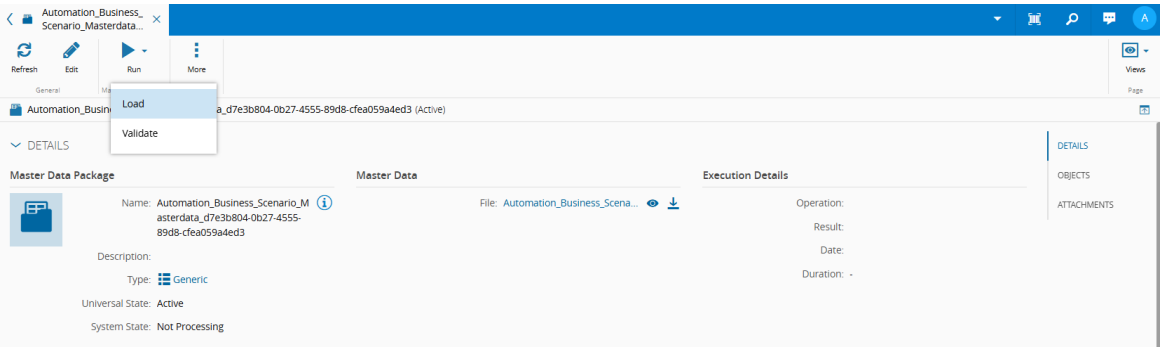
When you have selected the desired IoT Event Definitions, select `No` when prompted to add more IoT Event Definitions. The system then confirms if it should create the Master Data Package. Choose `Yes`.



Once created, the system sends the Master Data Package and you can access it directly from the link provided.

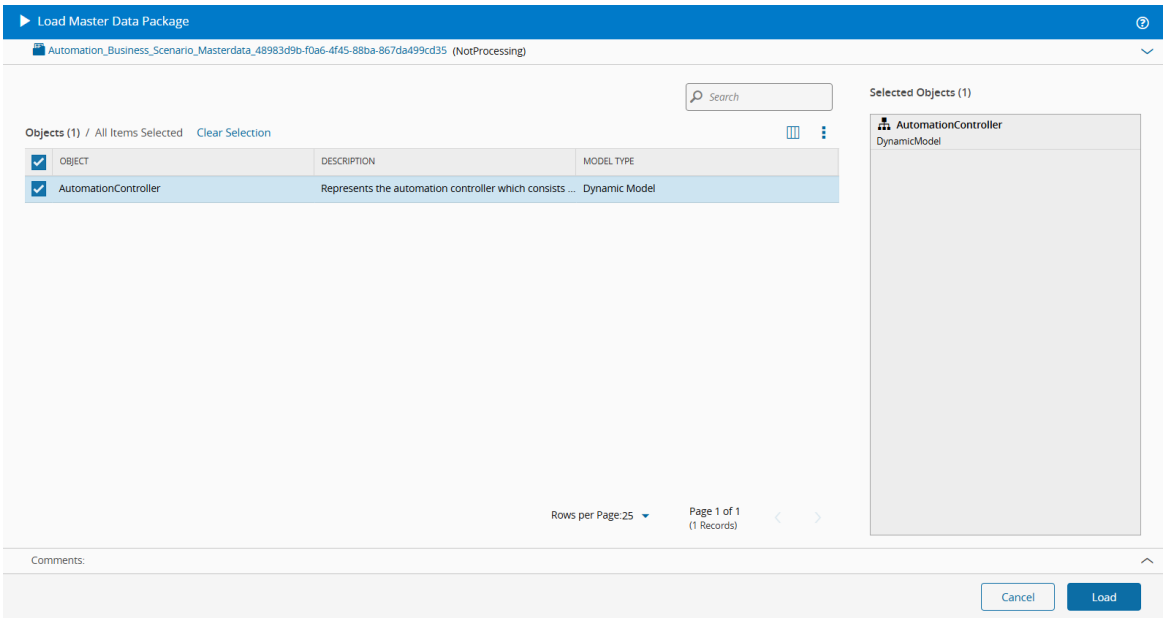
Sending Data to MQTT Broker

The next step is to load the master data package created. To do so, on the top ribbon, select **Run** and then **Load**.

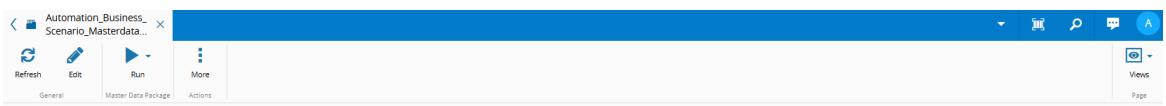


Then, select the `AutomationController` object and select the **Load** button. Once the load is complete, the system displays a success message.

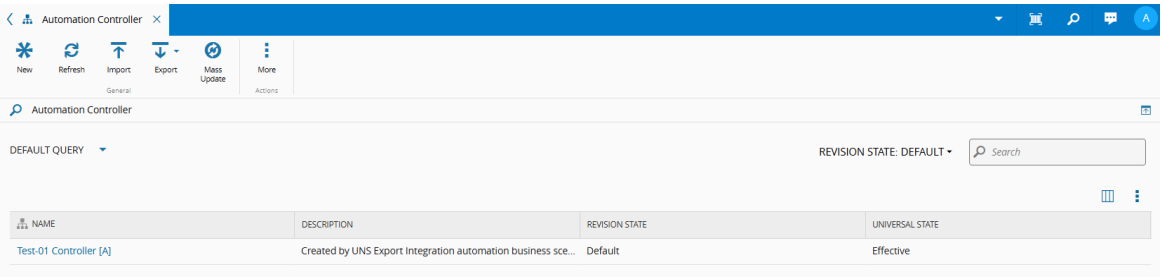
Loading Master Data Package



Successful Loading

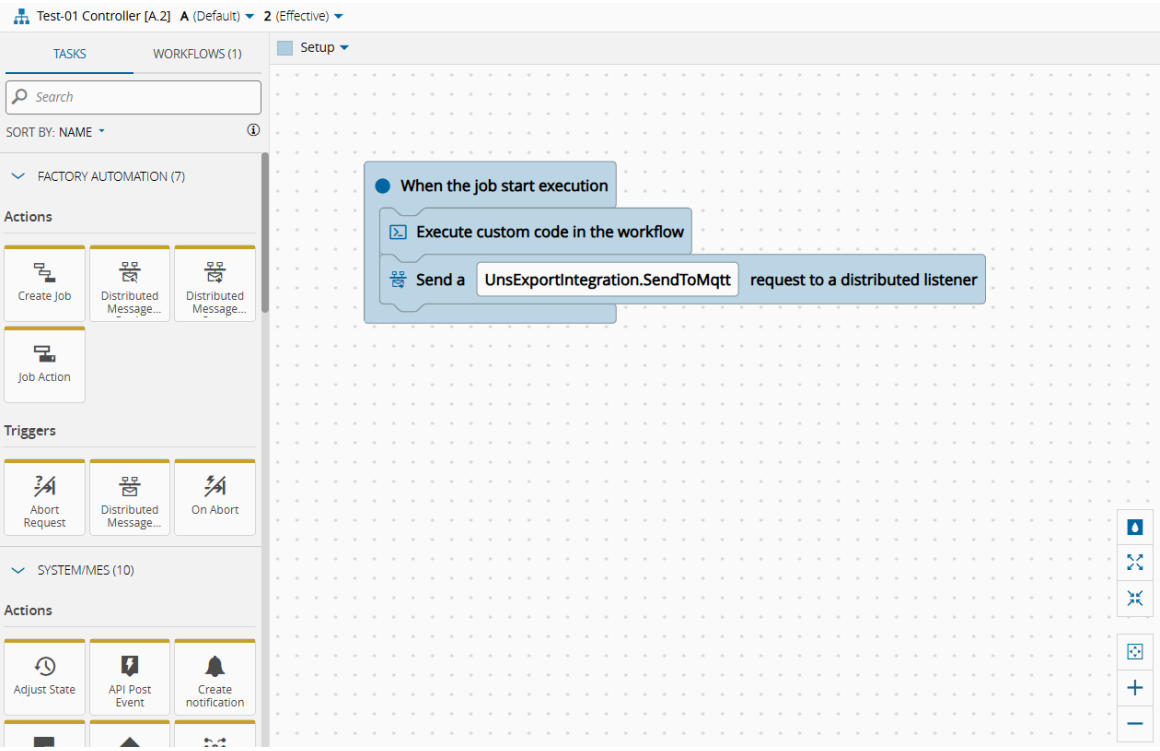


If you navigate to the **Data Platform Workflows** entity under the **Data Platform** menu, you will see the object created with the Master Data Package.

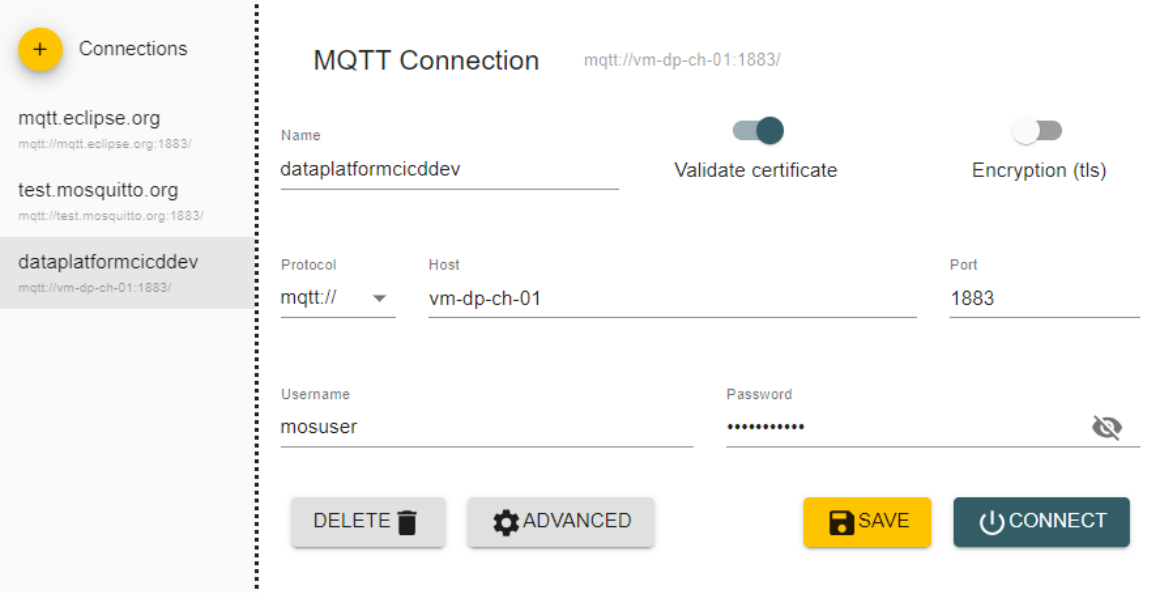


After the Master Data Package is loaded, the Automation Controller (`Test-01 Controller`) is ready to execute its workflow. The workflow follows three main steps:

- **Trigger on job execution** - starts automatically when a configured job begins execution.
- **Execute custom code** - passes the raw CDM event and transforms it into a valid MQTT topic.
- **Send to MQTT** - publishes the event using the `UnsExportIntegration.SendToMqtt` action, which forward the payload to a distributed listener connected to the MQTT broker.

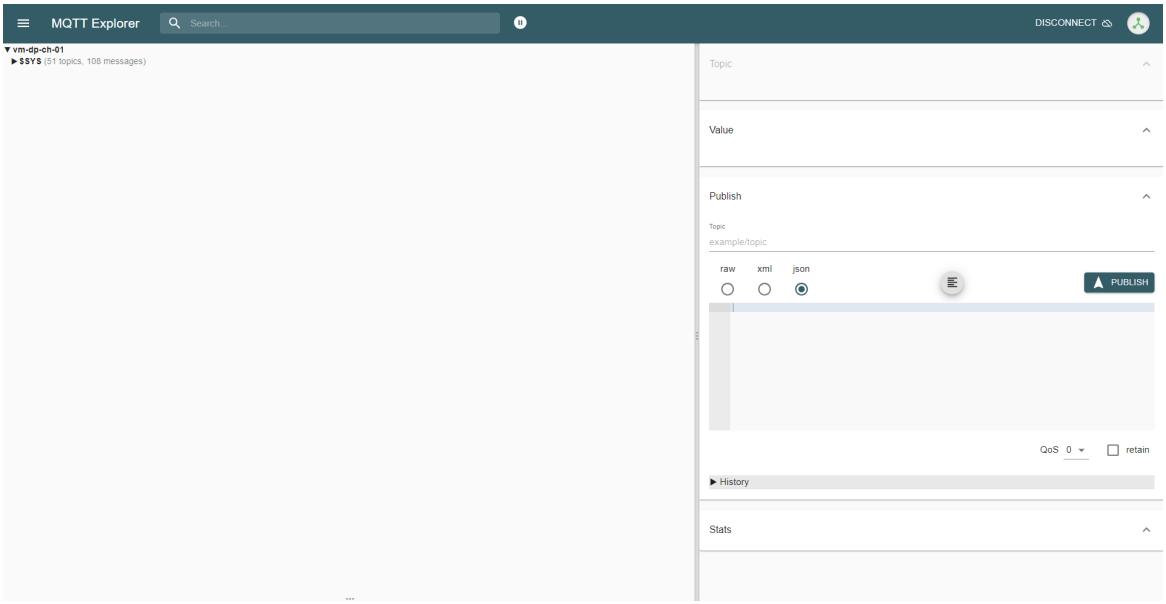


To confirm that events are reaching the broker, connect with an MQTT client such as MQTT Explorer. Use the same host address and credentials configured during the MES environment installation.

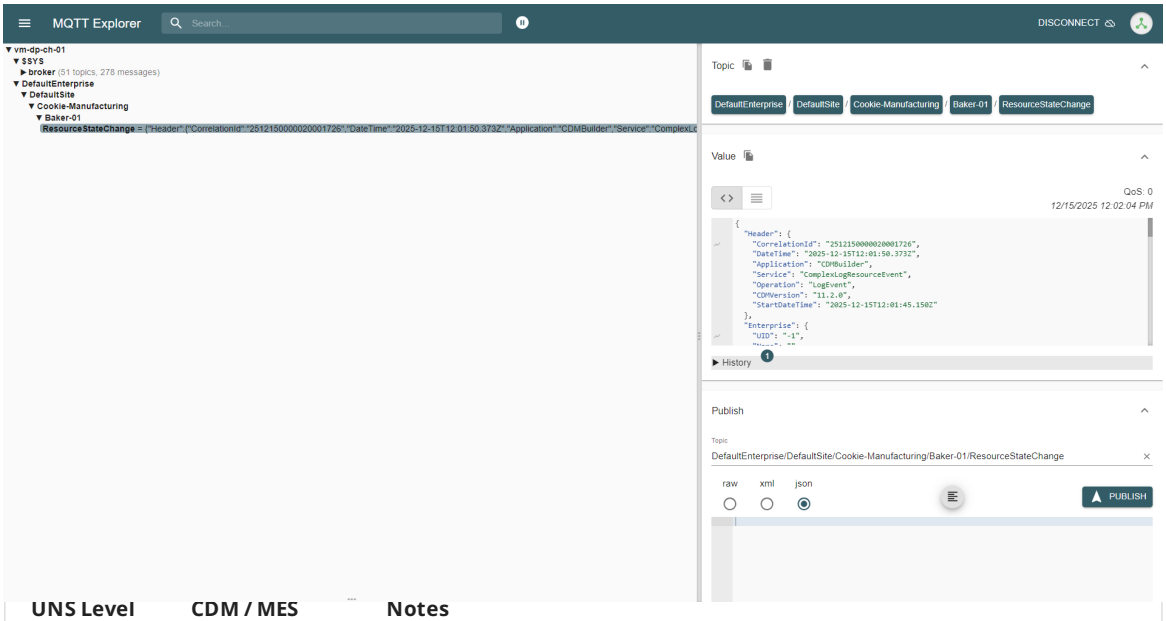


Perform a material operation or a resource state change - the two event types subscribed in this example. For instance, if the state of `Baker-01` changes from `Standby` to `Productive`, the update is propagated to the MQTT broker within seconds.

MQTT Before



MQTT After



UNS Level	CDM / MES Source	Notes
Enterprise	Enterprise entity	If no Enterprise is associated, `DefaultEnterprise` is used.
Site	Site entity	If no Site is associated, `DefaultSite` is used.
Area	Area entity	Represents the logical production area.
Resource	Resource entity	Typically maps to what ISA-95 refers to as a Work Center, depending on how resources are modeled in the MES.
Event	CDM Event Type	For example, `ResourceStateChange` or `MaterialOperations`.

Table: Mapping CDM Entities to UNS Levels

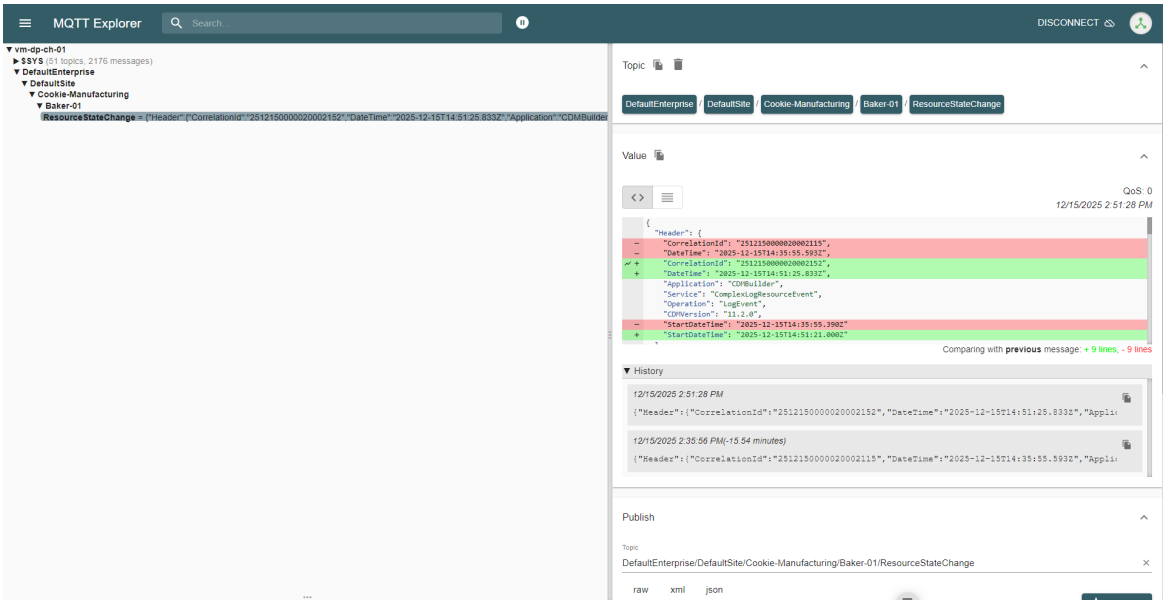
When the state of the resource `Baker-01` changes from `Standby` to `Productive`, a `ResourceStateChange` event is published to the corresponding UNS topic, following this structure: `DefaultEnterprise/DefaultSite/Cookie Manufacturing/Baker-01/ResourceStateChange`. And the MQTT broker receives the following JSON payload:

```
{
  "Header": {
    "CorrelationId": "2512150000020002115",
    "DateTime": "2025-12-15T14:35:55.593Z",
    "Application": "CDMBuilder",
    "Service": "ComplexLogResourceEvent",
    "Operation": "LogEvent",
    "CDMVersion": "11.2.0",
    "StartDateTime": "2025-12-15T14:35:55.390Z"
  },
  "Enterprise": {
```

```
    "UID": "-1",
    "Name": "",
    "AdditionalProperties": null
  },
  "Site": {
    "UID": "-1",
    "Name": "",
    "AdditionalProperties": null
  },
  "Facility": {
    "UID": "2512051538210000002",
    "Name": "Cookie Factory",
    "AdditionalProperties": null
  },
  "Area": {
    "UID": "2512051538210000001",
    "Name": "Cookie Manufacturing",
    "Type": "Cookie Production",
    "AdditionalProperties": null
  },
  "Resource": {
    "UID": "2512051538210000011",
    "Name": "Baker-01",
    "ResourceType": "General",
    "Model": "General",
    "Description": "Baker-01 Resource",
    "UniversalState": "Active",
    "Type": "Standard",
    "ParentResources": [],
    "OpenCorrectiveMAOsCount": 0,
    "Lanes": [],
    "Manufacturer": "General",
    "Class": "Process",
    "SystemState": "Up",
    "StateNames": [
      "Basic",
      "SEMI-E10"
    ],
    "StateValues": [
      "Up",
      "Productive"
    ],
    "SEMIE10State": "Productive",
    "CreatedOn": "2025-12-05T15:42:50.163Z",
    "CreatedBy": "2512051535120000002",
    "ModifiedOn": "2025-12-15T14:35:55.413Z",
    "ModifiedBy": "2512051535120000001",
    "ControlState": "Off-Line",
    "AdditionalProperties": null
  },
  "Previous": {
    "Resource": {
      "SystemState": "Up",
      "StateNames": [
        "Basic",
        "SEMI-E10"
      ],
      "StateValues": [
        "Up",
        "Standby"
      ],
      "SEMIE10State": "Standby",
      "LastModifiedOn": "2025-12-15T14:34:58.883Z",
      "LastModifiedBy": "2512051535120000001",
      "AdditionalProperties": null
    },
    "Area": {
      "UID": "2512051538210000001",
      "Name": "Cookie Manufacturing",
      "Type": "Cookie Production",
      "AdditionalProperties": null
    }
  }
}
```

```
}
```

If the state of `Baker-01` is later changed from `Productive` back to `Standby`, a new event is published to the same topic. MQTT clients such as MQTT Explorer visually highlight the differences between consecutive messages, as shown below.



So, in the **Value** section of the MQTT broker shown in the image above, the payload includes both the **current state** (highlighted in green) and the **previous state** (highlighted in red). This allows consumers to detect changes without maintaining local state, while each topic continues to receive new messages as events occur on the shop floor.

The **History** view in the MQTT broker confirms that all events are retained and ordered, providing:

- Full traceability of shop-floor changes;
- Real-time visibility across systems;
- A reliable event stream for downstream consumers.

This confirms that CDM events are successfully published through the UNS into the MQTT broker, enabling scalable, real-time insights across the shop floor.

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.