APPLICATION EXAMPLE

WinCC OA IT/OT Layer

SIMATIC WinCC Open Architecture

SIEMENS

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics, and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness, and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury, or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines, and networks.
In order to protect plants, systems, machines, and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines, and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g., firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/cert.

Contents

1. Introduction

1.1. Document Overview

This document provides a comprehensive overview of the WinCC OA IT/OT Layer Application Example, detailing its system architecture, configuration setup, and implementation process. It is designed to guide users through understanding the underlying architecture that enables the integration between IT and OT layers and how to configure the required parameters and data sources.

1.2. Components used

This application example has been created with the following hard- and software components:

Component	Number	Article number
WinCC OA 3.20 Server Basis	1	6AV6355-1AA50-0BA0
WinCC OA V3.20, Para Standard	1	6AV6355-1AA50-0CH0

Table 1-1 WinCC OA Licenses

You can purchase these components from the Siemens Industry Mall.

1.3. Requirements

- Linux operating system
- WinCC OA 3.20 Installation

2. Architecture

2.1. Project Architecture

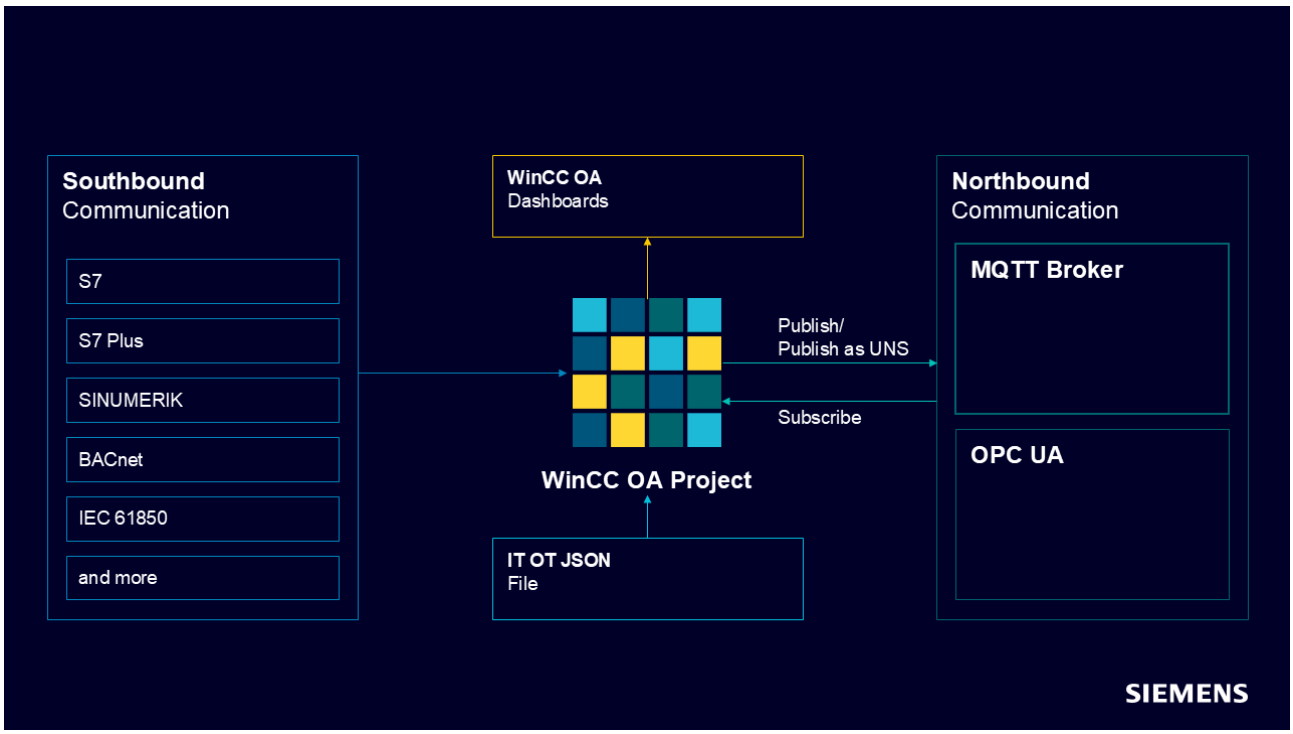

Figure *1*: IT OT Layer Architecture

The previous Figure 1 illustrates the architecture on which this application example is based on:
Once the project starts, it will load a JSON file located in the project directory, and out of this JSON file, the following configurations will be done automatically:

- All drivers and data points defined in the IT OT JSON file's "Southbound" section are created, including their corresponding PLC data addresses and configs.
- The required data points, as specified in the IT OT file, are published to the southbound layer, for example: MQTT brokers/OPC UA will be loaded from IT OT JSON file's "Publish" section
- To publish your data in the UNS (Unified Name Space) form, defining how your UNS will look like is taken from the IT OT JSON file's "UnifiedNameSpace" section
- Not only publishing to the southbound layer, but also subscribing to MQTT and OPC UA will be taken and configured and the datapoints will be created from IT OT JSON file's "Subscribe" section

  This use case is useful for scenarios like integrating data from MES (Manufacturing Execution Systems) or other external IT systems.
- Dashboards are automatically generated to visualize the data points specified in the IT OT JSON file's "Dashboard" section. These dashboards provide a real-time view of the system's key metrics and statuses.

2.2. IT OT JSON File Architecture

The file ITOTFile.json is a central configuration file located within the demo project at the path:
IT_OT_Demo\data\ITOTFile.json.
This file is structured into several key sections, each defining different aspects of the IT/OT integration. The file uses JSON format for readability and flexibility. Below is an overview of each section:

- **SystemConfig**
  Contains general metadata about the file, such as "version"
- **Southbound**
  Defines all the southbound tags—these are data points coming from the field via drivers (i.e., PLCs). This section maps the driver-level data sources and their configuration.
- **UnifiedNameSpace**
  Defines the hierarchical naming structure of presenting the datapoints.
  This section outlines how the namespace is organized, ending at the individual datapoint name (e.g., Factory/Line1/Machine1/TemperatureDataPoint).

  And this UNS path will be achieved in the **UnifiedNameSpace** section of the JSON file by writing it in a JSON form like what described in 2.2.3. "Southbound Section" This will be used during publishing to Northbound
- **Publish**
  Specifies which data points from the southbound section will be published to:
  - External MQTT
  - External OPC UA
  - Another example is Siemens Edge Data Bus broker
- **Subscribe**
  Describes the external data sources that will be **subscribed to**, including:
  - MQTT topics
  - OPC UA nodes
    This enables the demo to consume data from IT systems like MES or third-party services as an example of the use case.
- **Dashboard**
  Specifies the dashboards to be generated by the system. Each dashboard definition includes:
  - Dashboard name
  - Widgets to be displayed
  - Linked data points for real-time visualization

Each of these sections is explained in detail in the following subsections, including structure examples and detailed descriptions.

2.2.1. “SystemConfig” section

The SystemConfig section of the ITOTFile.json file is a flexible and open-ended configuration block. It is not processed by business logic, meaning any keys defined in this section are purely informational and used for metadata or documentation purposes.
This allows project teams to include relevant contextual information about the file without affecting the behavior of the demo.

Current Keys in Use:

- **version**:
  Specifies the version number of the IT OT file. This can be used for tracking updates and maintaining version control.
- **createdAt**:
  Indicates the creation date of the file. Useful for logging and file lifecycle tracking.
- **author**:
  Identifies the person or team who created or last modified the file.

Example:

```
"SystemConfig": {
   "version": "1.0",
   "createdAt": "2024-10-01",
   "author": "Ahmed Fadl"
 }
```

Since this section is not parsed by the demo's logic, additional keys can be added as needed.

2.2.2. “Southbound” section

Introduction:

- The Southbound section of the ITOTFile.json file defines all the field/device connections that supply real-time data to the WinCC OA project as an array of drivers.
- Each entry inside this array represents one southbound connection to a specific device or PLC
- The section is processed by the system’s logic: every connection, protocol configuration, and datapoint directly affects how data is read and archived.
- This enables you to configure, and manage different protocols (e.g., S7, SINUMERIK, BACNET, IEC61850) in one centralized place.

Structure & Purpose

Each object inside the Southbound array contains:

- **name**: A unique identifier for the connection (e.g., mt_S7Plus_11, sinumeriktestdriver).
- **description**: Free-text field to document the purpose of the connection.
- **protocol**: Specifies the communication protocol used with the field device (e.g., S7PLUS, SINUMERIK, S7, BACNET, IEC61850).
- **readCycle**: Defines how often (in seconds) data is polled from the device.
- **protocolData**: Holds protocol-specific parameters such as IP addresses,**and for other protocols** device IDs, or special protocol attributes (e.g., apName, sclFile and serverName for IEC61850) are needed

- **dataPoints**: An Array of datapoints to read from the connected device. And the following is more details about the possible content of the "**dataPoints"**

dataPoints

Each data point object inside a connection provides detailed configuration for a single sensor or signal:

- **name**: Unique name of the datapoint Element (e.g., LiquidTemperatureValue, FlowSetPoint).
- **description**: A field for additional description for the datapoint.
- **unit**: Engineering unit of the measured value (e.g., kg, ms, °C, or left **empty string**).
- **dataType**: Defines the type of the value (INT, DOUBLE, BOOLEAN, STRING).
- **archive** ***(W****here needed for the datapoint element)*: Is a **b**oolean flag that indicates if the value is to be stored historically.
- **minimumValue / maximumValue** *(W**here needed for the datapoint element)*: Allowing a WinCC OA range for the datapoint value.
- **dataPointData**:
  - **address**: Protocol-specific reference to the memory location (e.g., PLC DB address or IEC61850 reference).
  - **addressActive** *(where present)*: Indicates whether the address is currently active.
  - **onChange**: Whether the point triggers only when its value changes instead of each polling cycle.
  - **direction**: Defines how the value is acquired (e.g., READ), and in our IT OT use case in the Application example, the data is coming from the PLCs to the southbound.

Example

```
"Southbound": [
 {
  "name": "mt_S7Plus_11",
  "protocol": "S7PLUS",
  "readCycleInSeconds": "1",
  "protocolData": { "ipAddress": "158.226.215.241" },
  "dataPoints": [
   {
    "name": "LiquidTemperatureValue",
    "unit": "kg",
    "dataType": "INT",
    "archive": true,
    "minimumValue": 16,
    "maximumValue": 20,
    "dataPointData": {
     "address": "DB1001.DBD60",
     "addressActive": false,
     "onChange": false,
     "direction": "READ"
    }
   }
  ]
 }
```

]

Key Notes

- **Multiple Protocol Support**: Currently, the implemented protocols for the southbound communication are S7PLUS, SINUMERIK, S7, BACNET, and IEC61850
- **Reference Configuration**: The ITOTFile.json in the application example's *data* folder contains a comprehensive list of all possible configurations for the implemented southbound drivers.
- **Extensibility**: New connections or data points can be added by inserting new objects into the array without impacting existing configurations, then these configurations will be loaded once the project started or the "IT_OT_BL.ctl" control manager is started in console
- **Operational Impact**: Unlike SystemConfig, this section is always mandatory to be there in the JSON file

2.2.3. "UnifiedNameSpace" section

Introduction

- The **UnifiedNameSpace** section of the ITOTFile.json file defines the hierarchical plant model that will be **published to the northbound layer**.
- This hierarchy is **flexible**: you can define any number of intermediate levels, and it is **not mandatory** to include all levels (Plant → Production → Area → Line → Station → Equipment).
- Each branch **must end with datapoints** that are from the datapoints defined in the *Southbound* section.

Structure & Purpose

- Hierarchy levels (Plant, Area, Line, etc.) are **optional and customizable**, allowing a structure that fits the real or logical plant layout.
- Final nodes contain **arrays of tags**—these are the actual datapoint element names (e.g., WaterTemperatureValuie, HeaterOn) used for northbound publishing

Example

```
"UnifiedNameSpace": {
 "Plant1": {
  "Production2": {
   "Area1": {
    "Line1": {
     "Station1": {
      "Conveyor1": ["WaterTemperatureValuie", "HeaterOn"]
     }
    }
   }
  }
 }
}
```

Key Notes

- **Datapoints Consistency**: Every datapoint must match a valid datapoint from the *Southbound* section.

- **Extensibility**: You can add, remove, or reorganize levels and equipment at any time; only the final datapoints are required, and you can replace them with others.
- **Relation to Publishing**: How the UnifiedNameSpace is used during publishing is described in the next section, "**Publish**".
- **Reference Configuration**:
  The ITOTFile.json in the application example's *data* folder contains a representative example of a complete **UnifiedNameSpace** hierarchy, showing how different plant levels and final datapoints can be configured.

2.2.4. "Publish" section

Introduction

- The Publish section of the "ITOTFile.json" file defines how process data from WinCC OA is exposed to external systems.
- It describes all outbound connections→e.g., MQTT brokers or an OPC UA server—that receive live data.
- The current implementation supports publishing to two different MQTT brokers simultaneously and acting as one OPC UA server to serve data to interested clients.

Structure & Purpose

Each object inside the **Publish** array represents one outbound connection and contains:

- connectionName: A unique name for the connection (e.g., ExternalMQTT, ExternalOPCUA, InternalMQTT).
- connectionType: Defines the publishing protocol (MQTT or OPCUA).
- broker / port (MQTT only): Target broker address and port.
- initialTopic (MQTT only): Optional root topic for MQTT messages.
- exposedViews: Lists the Southbound protocol views to publish (e.g., S7PLUS, SINUMERIK).
- useUnifiedNameSpace:
  - false: Data is published directly by Southbound view names (per device connection on shouthbound).
  - true: Data is additionally published using the hierarchical **UnifiedNameSpace** structure. As well as published by Southbound view names (per device connection on shouthbound).
- credentials (MQTT only): Username and password if authentication is required.

Example

```
"Publish": [
 {
  "connectionName": "ExternalMQTT",
  "connectionType": "MQTT",
  "broker": "192.168.1.96",
  "port": "31884",
  "exposedViews": ["S7PLUS","SINUMERIK"],
  "useUnifiedNameSpace": false
 }
]
```

Key Notes

- **Multiple Outputs**: Two MQTT brokers can be published to at the same time, and an OPC UA server can expose data for clients.
- **Unified Namespace Option**: Setting useUnifiedNameSpace to true publishes the data following the **UnifiedNameSpace** hierarchy; otherwise, data is published directly by device name configured in the "Southbound" section.
- **Security:** Security features such as authentication, encryption, and certificate-based validation are not yet implemented.
- **Reference Configuration**:
  In the ITOTFile.json (located in the application example's *data* folder), the **Publish** array provides a complete ready-to-use configuration showing how MQTT brokers and an OPC UA server can be defined with their exposed views, UnifiedNameSpace options, and connection parameters.

2.2.5. "Subscribe" section

Introduction

- The **Subscribe** section of the ITOTFile.json file defines inbound connections from higher communication (Northbound) used to receive data from external sources into the WinCC OA project.
- It supports **OPC UA** and **MQTT** connections, allowing subscription to the number of servers needed or brokers.
- Each subscription can define its own polling cycle and a list of data points to be read.

Structure & Purpose

Each object in the **Subscribe** array describes one inbound connection with fields such as:

- **connectionType**: OPCUA or MQTT.
- **server / broker / port**: Connection parameters.
- **credentials** *(MQTT only)*: Username and password (values can be kept empty "optional").
- **readCycleInSeconds** *(optional)*: General polling interval for the connection.
- It can be set per connection or per datapoint. Per-connection settings apply to all datapoints unless a datapoint has its own entry
- **dataPoints**: Individual datapoints with properties like name, dataType, active, archive, minimumValue, maximumValue, and dataPointData.

Example

```
"Subscribe": [
 {
  "connectionType": "OPCUA",
  "server": "192.168.216.128",
  "port": "34841",
  "dataPoints": [
```

```
    {
     "name": "shiftNumber",
     "dataType": "DOUBLE",
     "dataPointData": {
      "address": "ns=2;s=Plant1.Production2.Area2.Packaging.Wrapper1.SimulationData",
      "onChange": true
     }
    }
   ]
  }
]
```


Key Notes

- **Multiple Inputs**: Any number of **MQTT** and **OPC UA** connections can be configured and used simultaneously.
- **Unified Flexibility**: Each datapoint can have its own read cycle, activation flag, archive setting, and value range.
- **Security**: Security features—including **certificates are not yet implemented**.
- **Reference Configuration**:
  In the ITOTFile.json (located in the application example's *data* folder), the **Subscribe** array provides a complete ready-to-use configuration showing how multiple OPC UA and MQTT connections can be defined with their datapoints, polling cycles, and addressing (topics or OPC UA node addresses).

2.2.6. "Dashboard" section

Introduction

- The **Dashboard** section of the ITOTFile.json file defines the **operator dashboards** to be shown in WinCC OA.
- Each dashboard contains a set of **widgets** arranged in a grid (x/y position, rows, and columns) to visualize live data such as gauges, trends, pie charts, and labels.
- The widgets listed in the JSON are the **only supported widget types at present**. New widget types can be introduced by modifying the IT_OT_BL.ctl control manager.

Structure & Purpose

Each object inside the **Dashboard** array contains:

- **name**: Dashboard name.
- **description**: Text describing the dashboard purpose.
- **widgets**: An array of visual elements to display, where each widget includes:
  - **type**: Widget type (e.g., gauge, trend, pie, label).
  - **title**: Display title.
  - **x / y / rows / cols**: Position and size in the grid.
  - **dataPoint / dataPoints**: Associated datapoint elements(s) to be visualized based on the widget type.
- **dataPointsDescription:** The description of each datapoint that has been chosen with the widget

- **rangeSettings** *(where applicable per widget)*: Optional configuration for min/max values and range type (e.g., manual, oa).

Example

```
"Dashboard": [
 {
  "name": "Dashboard1",
  "description": "This is my dashboard description1",
  "widgets": [
   {
    "type": "gauge",
    "title": "TotalOEE",
    "x": 0,
    "y": 0,
    "rows": 9,
    "cols": 11,
    "dataPoint": "TotalOEE",
    "rangeSettings": {
     "type": "oa",
     "max": 100,
     "min": 9.0
    }
   }
  ]
 }
]
```

Key Notes

- **Widget Types**: The available widgets (gauge, trend, pie, label) are those currently implemented. Additional widget types can be added only by editing the **IT_OT_BL.ctl** control logic.
- **Reference Configuration**: The full dashboard setup, including all dashboards and widget definitions with the possible entries per widget, is found in the ITOTFile.json in the application example's *data* folder under the **Dashboard** array.

3. IT OT Project Operation

This chapter describes in detail how to operate the IT/OT demo project in WinCC OA. The purpose is to provide step-by-step guidance for launching the demo, monitoring its execution, and validating that all required datapoints, connections, and dashboards are correctly created and functional. Each figure corresponds to a specific step in the workflow and is referenced in the explanations below.

3.1. Starting the Project

ITOTFile.json content

Change the needed configuration in the IT OT file located in the project directory at: IT_OT_Demo/data/ITOTFile.json

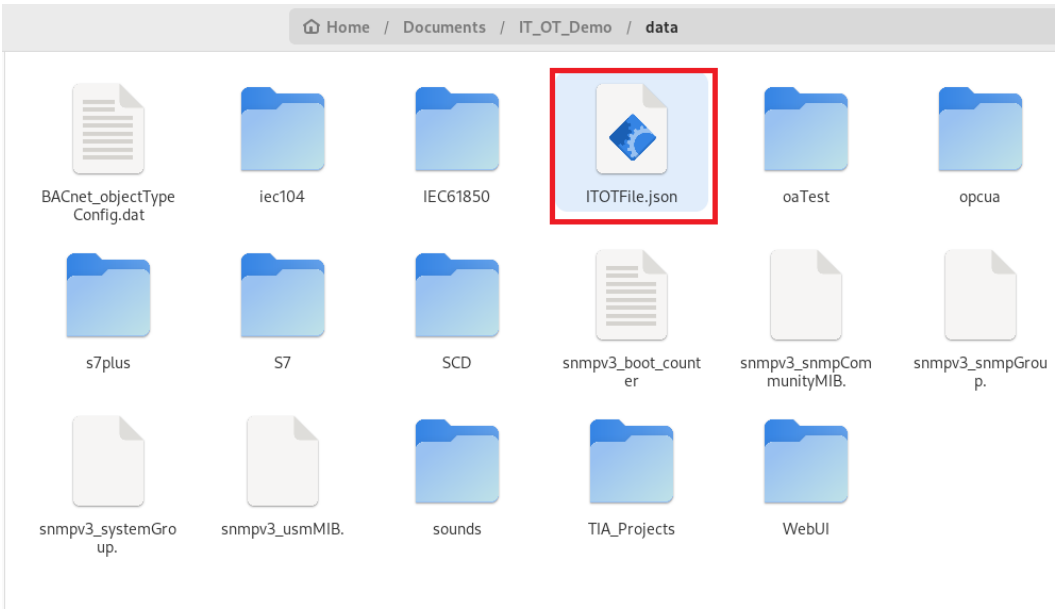

Figure 2: IT OT Json file navigation in the project subfolders

Starting the Project / Control Manager

To execute the demo logic, start the project IT_OT_Demo from the WinCC OA Console. If the project is already running, it is sufficient to start the dedicated Control Manager manually. Start the manager associated with the script file IT_OT_BL.ctl as shown in Figure 3

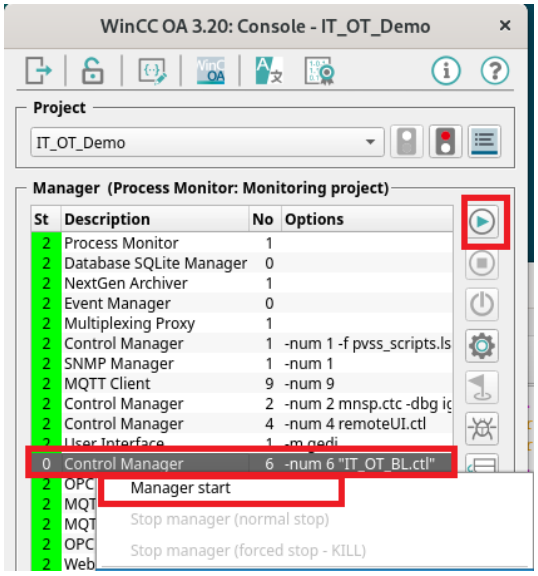

Figure 3: WinCC OA Console

Monitoring the Log Output

When the business logic starts, the system produces log entries reporting the important actions/results. These log entries can be monitored in the Log Viewer. All messages related to the IT/OT demo are tagged with the prefix “ITOT_LOG”. By filtering the log viewer for this keyword, you can immediately focus on the relevant events, warnings, and status updates of the demo execution as appeared in “Figure 4”.

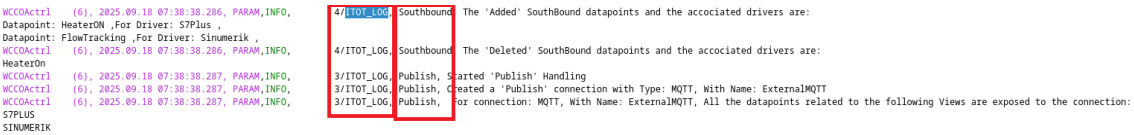

Figure 4: WinCC OA Log Viewer

Completion of Initialization

After the demo logic has completed its initialization and has finished setting up all datapoints, namespaces, and connections…ETC, the control manager that was running IT_OT_BL.ctl will stop automatically. This indicates that the configuration was successfully processed and that the system is ready for verification as shown in “Figure 5”

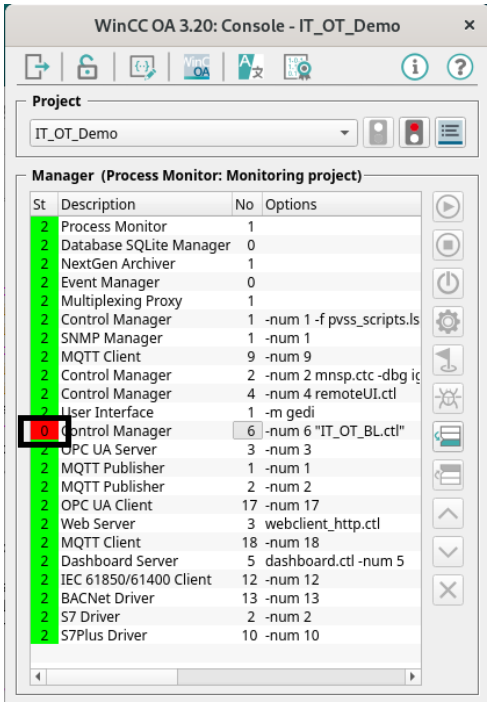

Figure 5: WinCC OA Console

3.2. Verifying Created Plant Model Editor Views (CNS Views)

Checking the CNS Viewer

The CNS Viewer (Plant Model Editor) can be opened by opening GEDI UI and by clicking the corresponding icon in the WinCC OA Console. This tool provides a structured view of all namespaces generated in the demo as appeared in “Figure 6”.

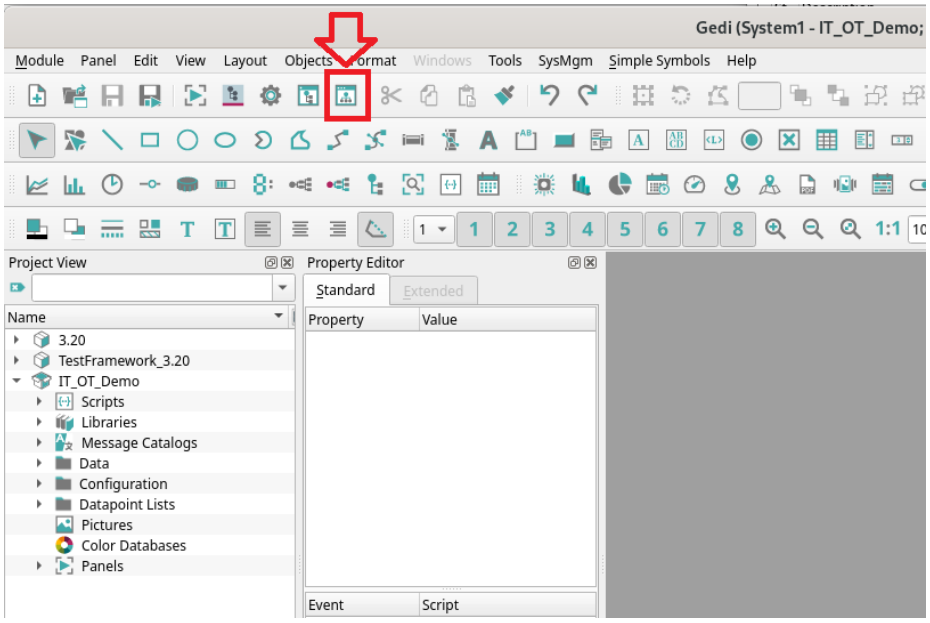

Figure 6: Plant Model Editor Icon

Unified Namespace and Device Data

Within the CNS Viewer, you can explore the automatically created device structures as well as Unified Namespace (UNS). This unified hierarchy shows how datapoints from different Southbound sources are normalized and aggregated as shown in “Figure 7”

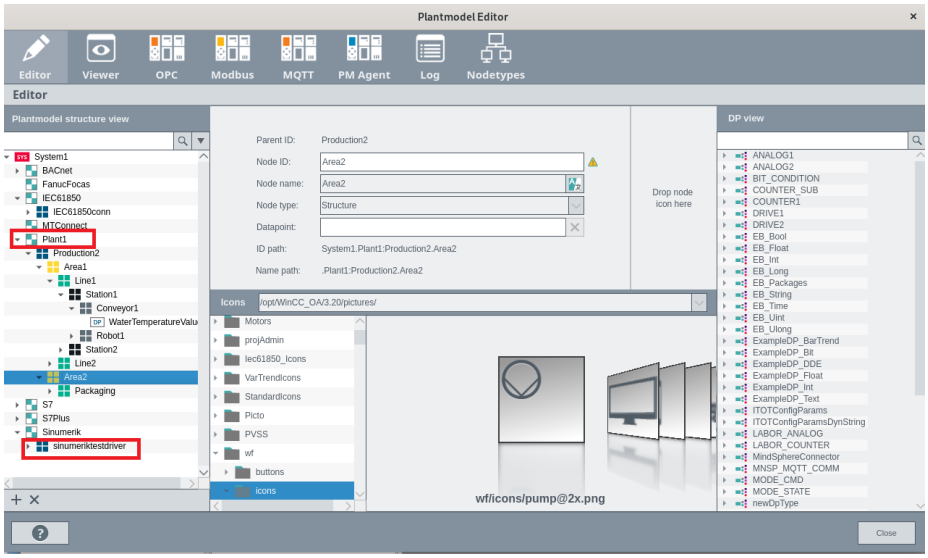

Figure 7: Plant Model Editor Panel

Southbound Datapoint Mapping

For each device, Southbound tags linked to datapoints can be inspected. The specific link is shown in the Datapoint field like shown in “Figure 8”

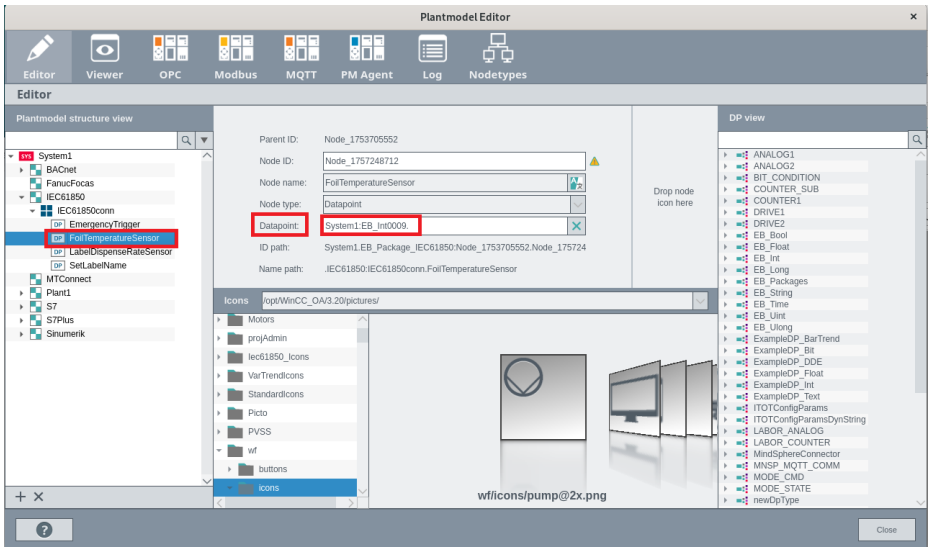

Figure 8: Checking the Southbound tags linked to datapoints in CNS

OPC UA Exposure

By switching to the OPC tab in the Plant Model Editor, the user can verify that the datapoints are exposed via the built-in OPC UA Server. The server structure includes both device-specific datapoints as well as the full UNS hierarchy like shown in “Figure 9”

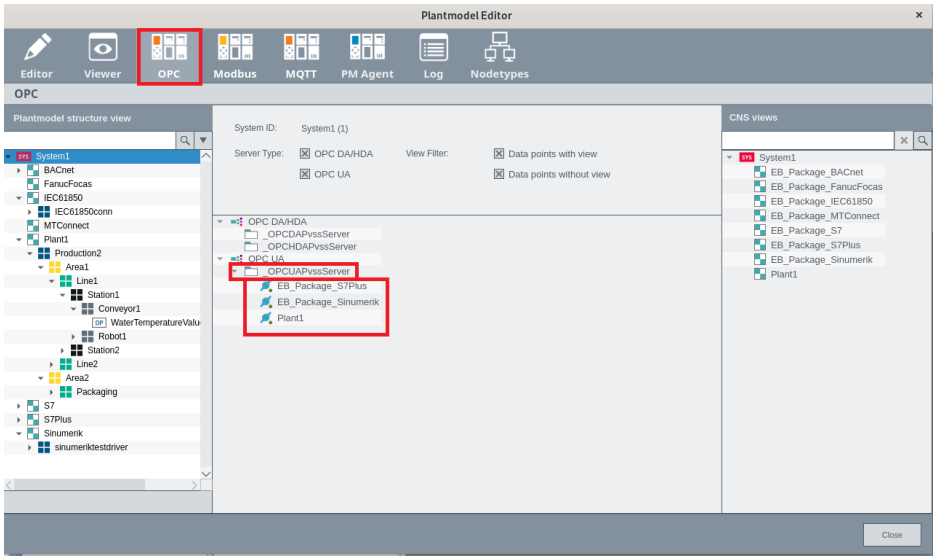

Figure 9: Checking the OPC UA Data Exposure in CNS

MQTT Exposure

Similarly, under the MQTT tab in the Plant Model Editor, the published topics can be inspected. Each configured MQTT Publisher exposes the datapoints per device and through the UNS like shown in “Figure 10”

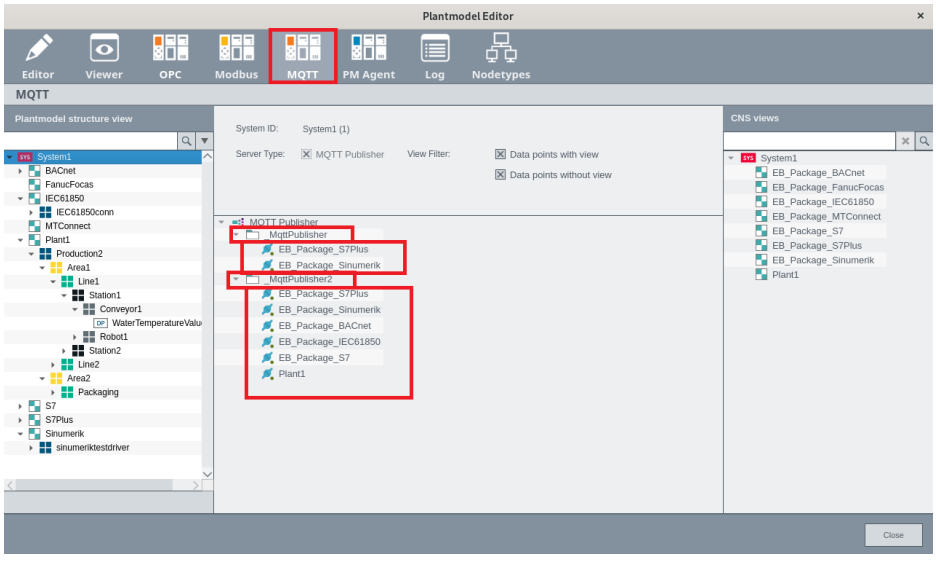

Figure 10: Checking the MQTT Data Exposure in CNS

3.3. Checking Datapoints and Connections

Southbound Datatypes

All Southbound datapoints created by the connected drivers are grouped under the “EB_*” datatypes. These contain the linked datapoints to the tag values that originate from PLCs like shown in “Figure 11”

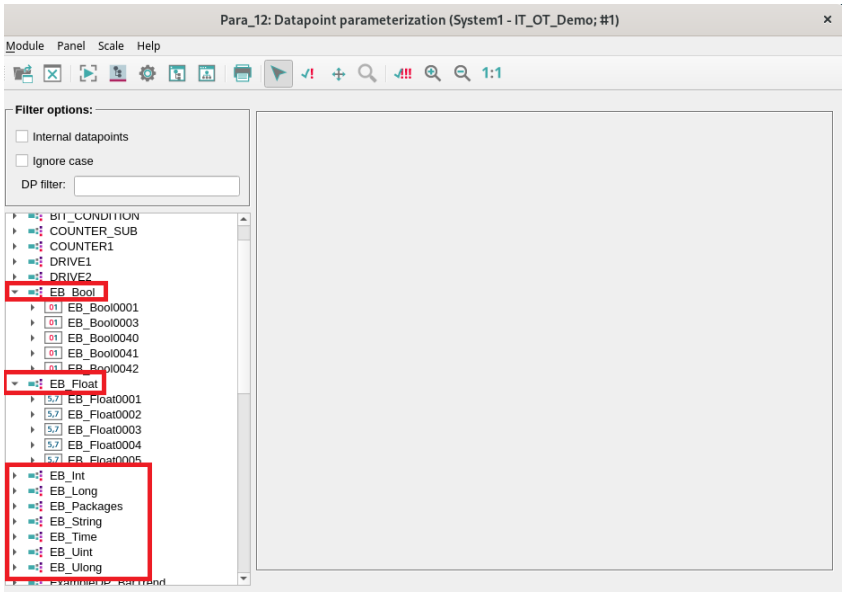

Figure 11: Para Southbound data point types

Subscribed Datatypes

Datapoints that are subscribed from IT systems via OPC UA or MQTT connections are placed under the SB_Sub_* datatypes as shown in “Figure 12”

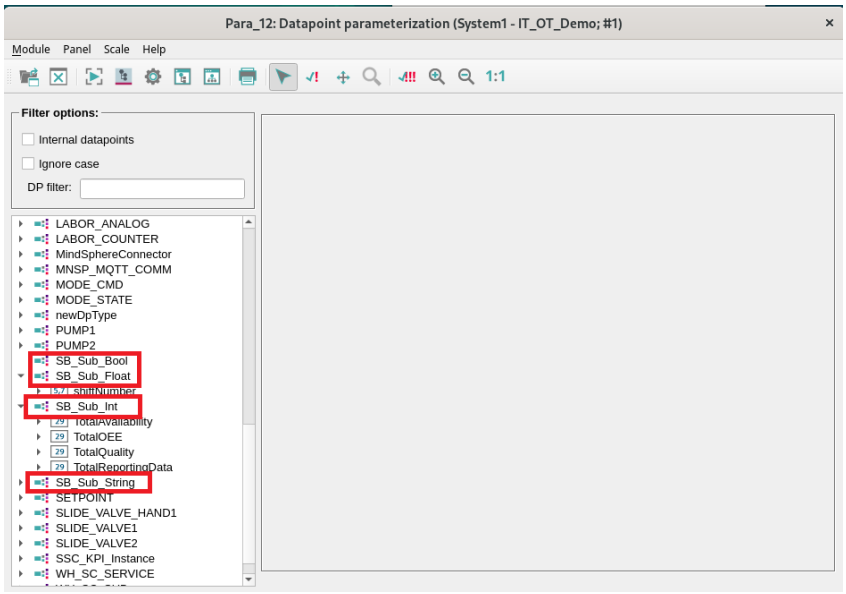

Figure 12: Para Northbound subscribed data point types

Northbound Subscribe Drivers Connection

To check the OPC UA and MQTT drivers' connection,

You can open (system management→Driver OPC→OPC UA Client) then check the OPC UA Driver connection parameters like shown in "Figure 13"

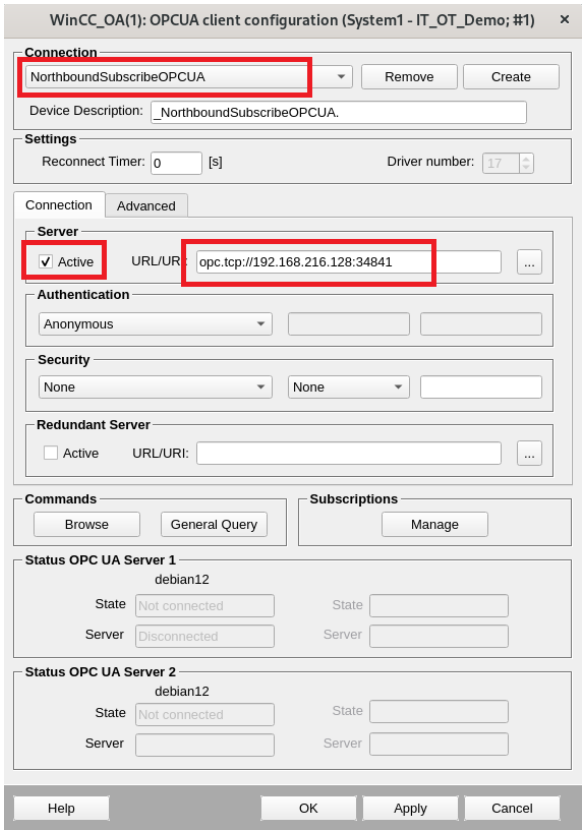

Figure 13: OPC UA Client Connection Parameters

And for MQTT driver, you can check it through (system management --> Driver --> MQTT Client) like shown in "Figure 14"

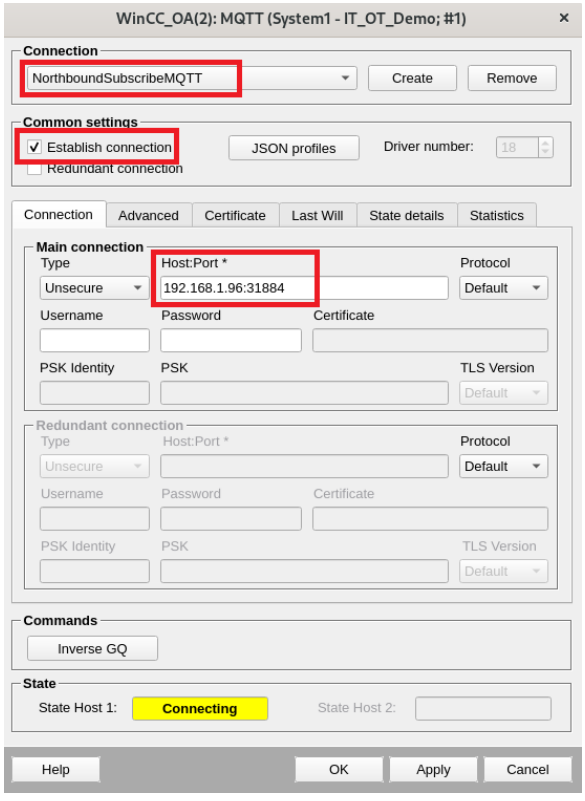

Figure 14: MQTT Client Connection Parameters

Southbound Drivers Connection

To check the southbound driver's connection,

We can check them through system management with the same way you checked the Northbound drivers

As an example: to check the S7+ driver connection, through (system management → Driver S7 → S7+ Driver) like shown in "Figure 15"

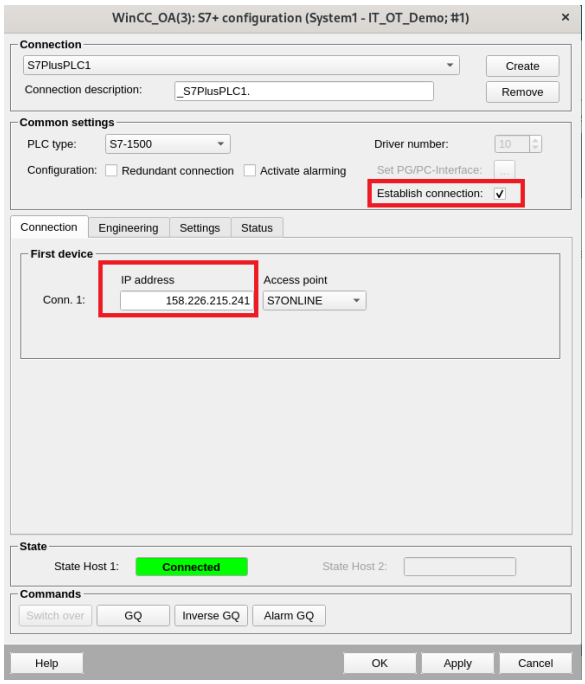

Figure 15: S7+ Connection Parameters

3.4. Validating Configuration Parameters

Address Mapping

In the datapoint parameterization view, the address settings can be inspected. This confirms that each datapoint is correctly mapped to its external address like shown in "Figure 16"

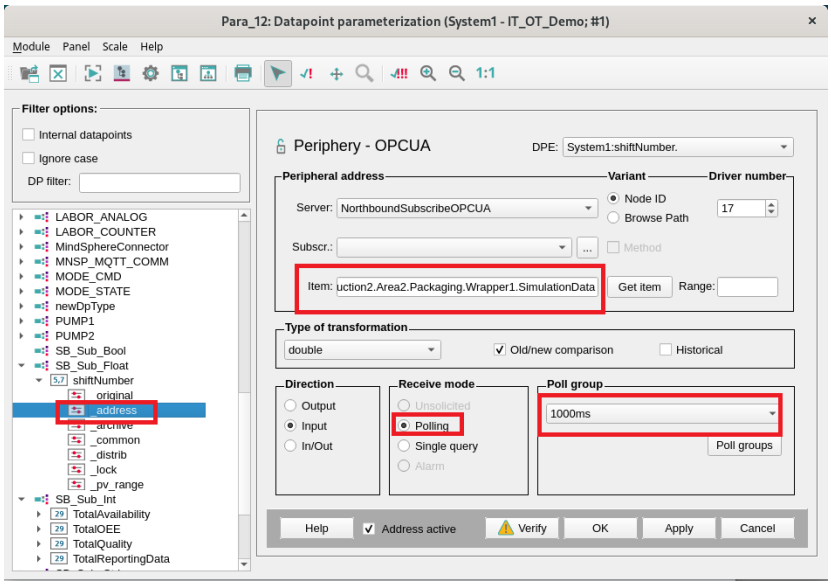

Figure 16: Datapoints Address Configs

Archive Configuration

The archive configs define which datapoints are stored historically or not, and this also can be checked if it is configured for the datapoint through PARA like shown in “Figure 17”

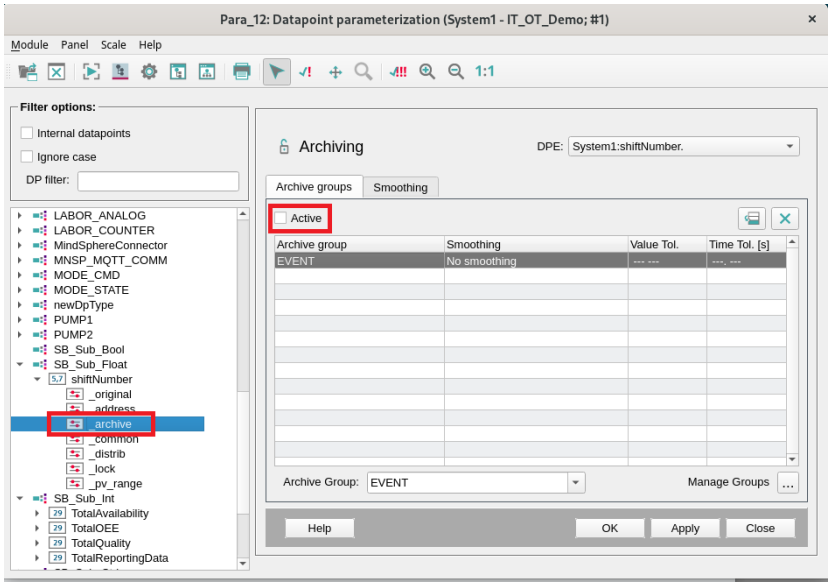

Figure 17: Datapoints Archive Config

Other Configuration

Also, the other configuration specified in the IT OT JSON file can be checked or validated through PARA, for example the “PV Range” config, like shown in “Figure 18”

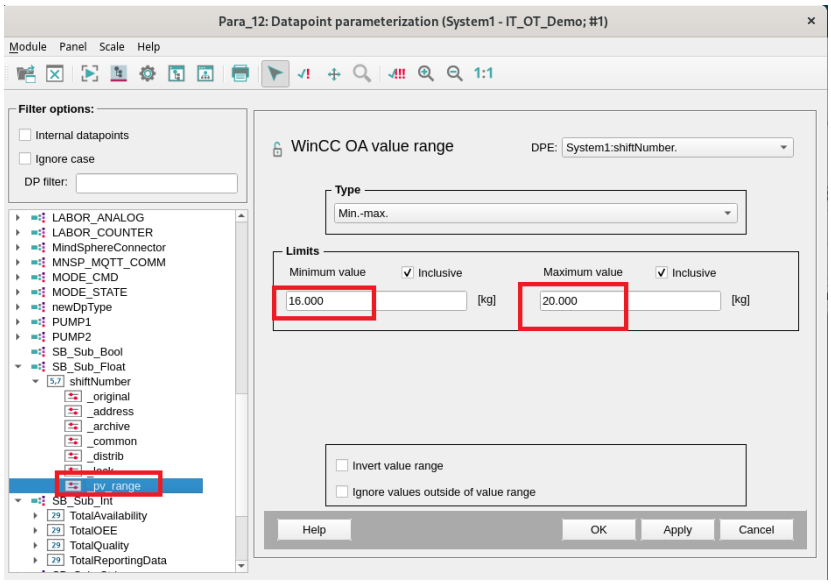

Figure 18: Datapoints PV Range Config

3.5. Dashboards

Dashboard Datapoints

Newly created dashboards are represented in the system under the "_Dashboard" datatype as datapoints like shown in "Figure 19"

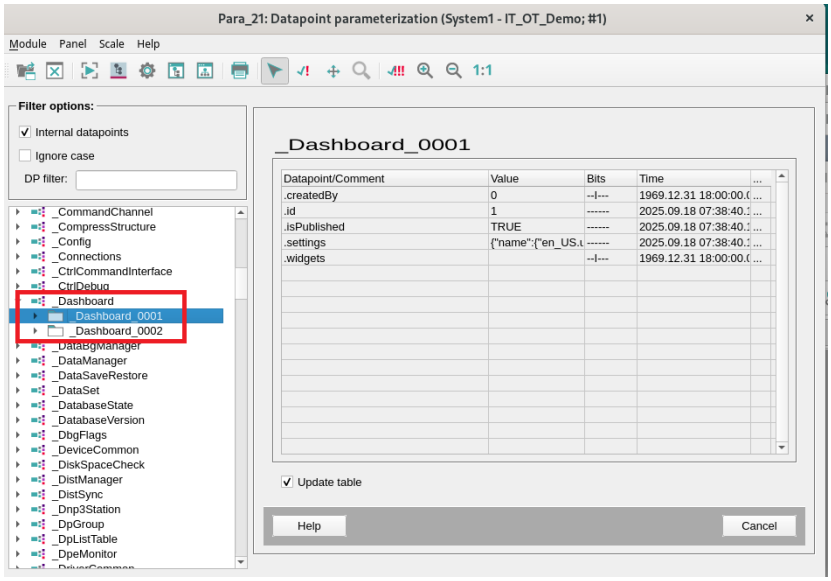

Figure 19: Dashboards datapoints

Dashboard Overview in Browser

The dashboards can be accessed through any browser at https://localhost:31090/.

Login with an already created user: "root2" password: "Siemens123!" or create a new user.

The overview page lists all available dashboards like shown in "Figure 20"

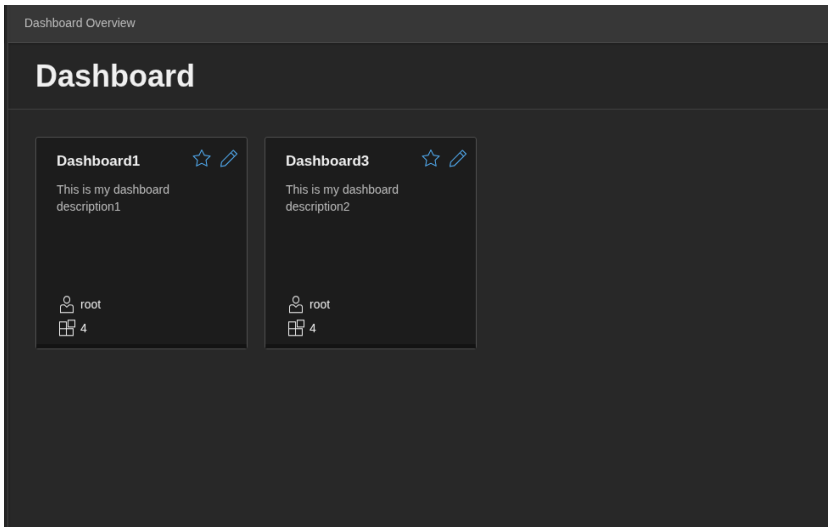

Figure 19: Dashboards overview

Live Dashboard View

Opening a dashboard shows the configured widgets such as gauges, labels, pie charts, and trends. Each widget is bound to a datapoint, and live values are updated automatically when the linked datapoints values are changed like shown in “Figure 21”

Figure 19: Dashboard Example

4. Appendix
Important WinCC OA specific abbreviations

Acronym	Long form	Meaning
WinCC OA	Simatic WinCC Open Architecture	A SCADA system for visualizing and operating of processes, production flows, machines and plants in all lines of business. Distributed systems enable any number of stand-alone systems, from 2 to 2048, to be linked via a network. Each subsystem can be configured either as a single-user or multi-user system, redundant or not, in each case.
DPT	Data point type	Object definition (class) of a structured data object as mapping of a real device. Single data points (instances) are derived from the DPT. Therefore, the data point type is a form of template.
DP	Data point	Structured, device-oriented data object as representation of a real device within the control system. A data point contains one or more data point elements (process variables).
DPE	Data point element	Single process information within a device-oriented data point. Each DPE corresponds to a value/state. In addition to the value, there are DPE attributes like time stamp, quality information or origin.
GEDI	Graphic Editor GEDI	Graphic Editor. It is used both for drawing of process images ("panels") as well as for designing of symbols, dialogs, and scripting.
PARA	Configuration Tool	Editor for the creation and configuration of data point types, data points, and data point elements as well as their configs.
ASCII	American Standard Code for Information Interchange	Standardized protocol for storage and transfer of characters/text. In WinCC OA, the acronym also refers to the database import/export manager. It is a module to export and import configurations as ASCII files. Mass configuration can therefore be carried out in a spread sheet program (for example MS Excel), file editor, or in an external database.
D	Driver Manager ("Driver")	Interface for connecting controllers (PLC, DDC, ...) fieldbuses and telecontrol systems. A driver handles the communication via an external protocol and enables the exchange of information with WinCC OA. The processing of data from the “field” to WinCC OA contains event orientation, old/new comparison, transformation, conversion, and smoothing. The protocol of the Driver must be the same as the protocol of the “field” device. Furthermore, the connection (how to reach the device) must be configured in WinCC OA. For exchanging data, a periphery address must be configured on a corresponding DPE.
CTRL	Control Manager ("Scripting")	Processing unit that allows to process user specific logic / business logic (control scripts). Control possesses an easy to learn syntax (similar to ANSI-C) and is processed by an interpreter (CTRL Manager).

Table 4-1 WinCC OA specific abbreviations

Service and support

WinCC OA Extended Services

Do you have questions about WinCC OA projects, need additional features, or require technical assistance? ETM provides 24/7 access to our complete service and support expertise for WinCC OA.

Our range of services includes the following:

- Extended Services provide tailored support for your evolving needs
- All kinds of analysis/troubleshooting for older WinCC OA versions than our current mainline
- Project startup workshop
- Architecture definition
- Project engineering assistance with dedicated contact person
- Special project developments (special requirements, web widgets, gateways, etc.)
- WinCC OA library development assistance
- Project or architecture reviews with report
- Project-specific problem or performance analysis
- Project upgrade (analysis with report, assistance during upgrade, etc.)
- Assistance for complex error reproduction scenarios
- On-site assistance for any tasks related to WinCC OA and their components
- 24/7 on-duty assistance or priority callback for certain time range
- Database support Oracle®, InfluxDB®, PostgreSQL® and MS SQL®
- Raima/HDB to SQLite/NGA migration
- Setup and consulting for WinCC OA Add-ons e.g., APM, AMS, DRS, …
- WinCC OA Security services (as per the WinCC OA Security Guideline, NIS2)
- Tests on unsupported platforms (e.g., unsupported OS)
- Individual Workshops (driver workshop, UI workshop, business logic, etc.)
- Factory Acceptance Test (FAT)/Site Acceptance Test (SAT) assistance
- Creating of prototypes, proof of concepts, demos, etc.
- Project tender analysis and evaluation of projects
- You can find detailed information on our range of services in the service catalog web page:

  www.winccoa.com/documentation/WinCCOA/latest/en_US/Support/topics/support_extendedServices.html

SiePortal

The integrated platform for product selection, purchasing and support - and connection of Industry Mall and Online support. The SiePortal home page replaces the previous home pages of the Industry Mall and the Online Support Portal and combines them.

- **Products & Services**
  In Products & Services, you can find all our offerings as previously available in Mall Catalog.
- **Support**
  In Support, you can find all information helpful for resolving technical issues with our products.
- **mySieportal**
  mySiePortal collects all your personal data and processes, from your account to current orders, service requests and more. You can only see the full range of functions here after you have logged in.

You can access SiePortal via this address:

sieportal.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers.
– ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

WinCC OA – Training and Certification

To fully leverage the flexibility and openness of WinCC OA, we offer a wide range of training courses, from beginner to expert levels. Our modules cover various topics, with options for individual training. For WinCC OA partners, completing specific courses is required to obtain or maintain partner status, ensuring the highest level of expertise and support.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

www.winccoa.com/support/training.html

Change documentation

Version	Date	Modifications
V1.0	10/2025	First version

Table 4-2 Change Documentation

Published by
Siemens AG
DI FA HMI ISW ETM
Marktstrasse 3
7000 Eisenstadt
Austria
E-mail: wincc_oa.at@siemens.com
Web: www.siemens.com/wincc-open-architecure

For the U.S. published by
Siemens Industry Inc

100 Technology Drive
Alpharetta, GA 30005
United States