AMORPH.pro
SMARTUNIFIER

Send file (XML) and database data via MQTT

Release 1.0.0

Amorph Systems GmbH

Nov 25, 2021

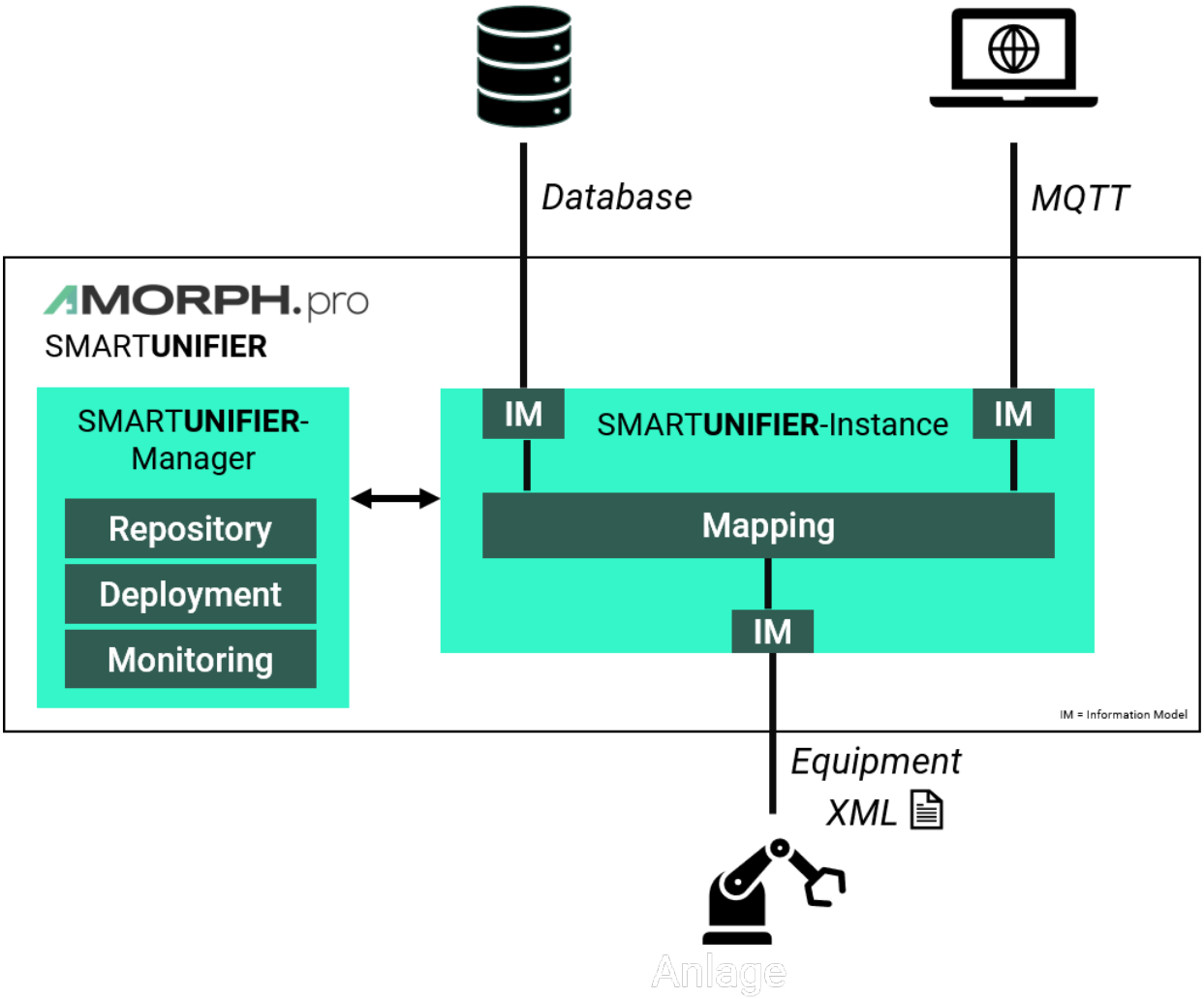
CONTENTS

CHAPTER

ONE

OVERVIEW

This Scenario describes step by step how XML data can be sent via MQTT enriched with additional data from a database. This scenario shows also how type conversion and date formatting can be implemented via the SMART**UNIFIER** Mapping.



CHAPTER

TWO

PREREQUISITES

1. Equipment Data - (XML file)

```
<?xml version="1.0" encoding="utf-8"?>
<ProductionResult>
   <OrderNumber>PO_000001</OrderNumber>
   <ProductNumber>F2PZJ55QW11</ProductNumber>
       <Date>2021-03-31T07:20:41.214Z</Date>
   <Quality>IO</Quality>
   <Quantity>5</Quantity>
</ProductionResult>
```

2. SQL Server (Database)

Create Table

```
create table DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER (MAIN_KEY bigint IDENTITY(1,1) PRIMARY KEY, CUSTOMER_NAME nvarchar(max), ORDER_NUMBER nvarchar(max))
```

Insert Data

```
INSERT INTO DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER (CUSTOMER_NAME, ORDER_NUMBER) VALUES ('DemoCompany1', 'PO_000001'); INSERT INTO DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER (CUSTOMER_NAME, ORDER_NUMBER) VALUES ('DemoCompany1', 'PO_000002'); INSERT INTO DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER (CUSTOMER_NAME, ORDER_NUMBER) VALUES ('DemoCompany2', 'PO_000003'); INSERT INTO DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER (CUSTOMER_NAME, ORDER_NUMBER) VALUES ('DemoCompany3', 'PO_000004');
```

3. MQTT Client (For testing)

Download the MQTT Explorer.

CHAPTER

THREE

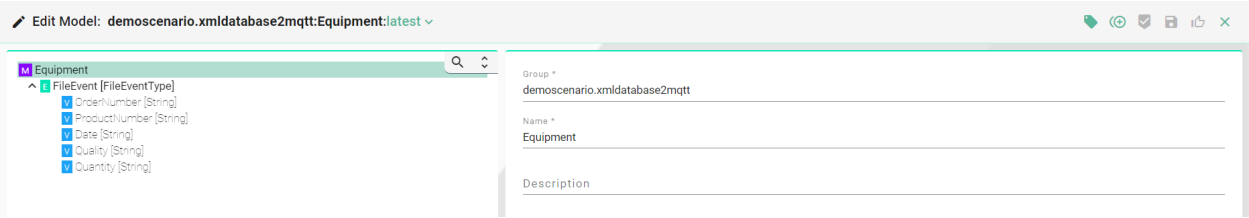
INFORMATION MODEL

3.1 Information Model - Equipment

Create an Information Model that represents the structure of the XML-file.

Structure of the *XML* - Information Model:

- Event that represents the **trigger** for the Mapping. If a new file is recognized by SMART**UNIFIER** the Rule in the Mapping will be executed.
- Variables (of data type String) under the Event represent the key-value pairs from the XML-file

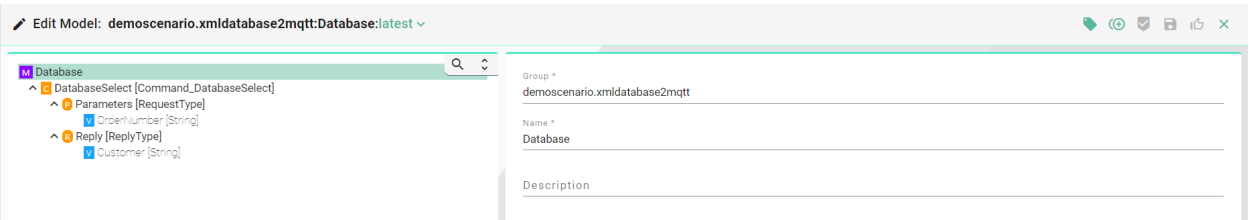


3.2 Information Model - Database

Create an Information Model, which will be used for the **Select** query.

Structure of the *Database* - Information Model:

- Command which is executed once the trigger is activated.
- Parameter variable **OrderNumber** (of data type String) that is used in the SELECT query later on.
- Reply variable **Customer** (of data type String) that holds the result of the query.

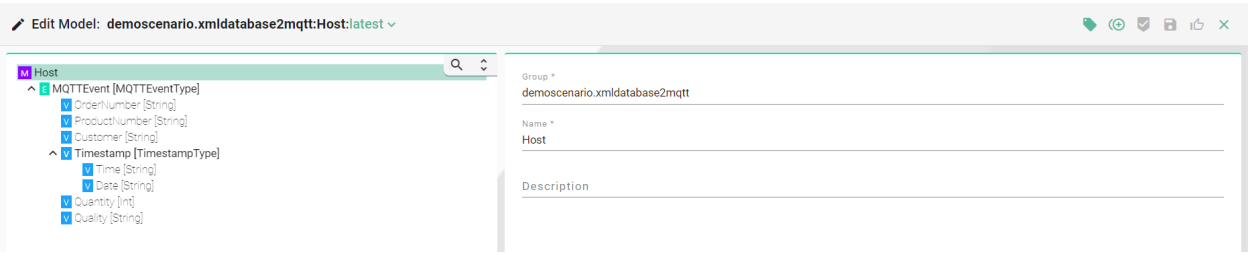


3.3 Information Model - Host (MQTT)

Create an Information Model that represents the structure of the Host (in this case MQTT).

Structure of the *MQTT* - Information Model:

- Event which is used to trigger the transfer of the data.
- Variables:
  - OrderNumber, ProductNumber, Customer, Quality - of type `String`.
  - Quantity of type `Int`
  - Timestamp (custom data type)
    - date, time - of type `String`



CHAPTER

FOUR

COMMUNICATION CHANNEL

4.1 Communication Channel - Equipment

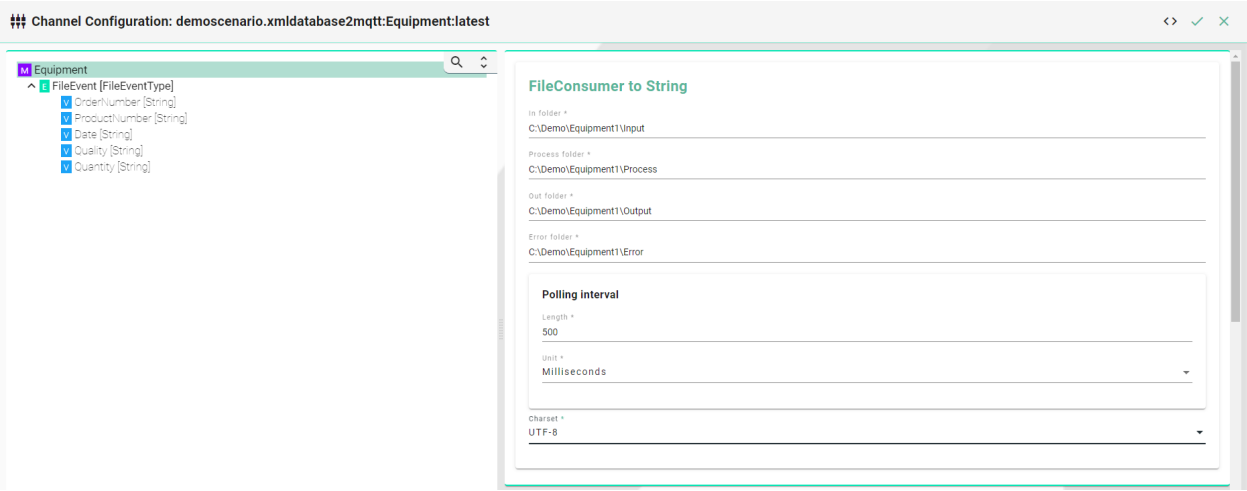
In this scenario the XML-file is processed by the SMART**UNIFIER** with the build-in File Reader.

1. Create File Reader Channel:

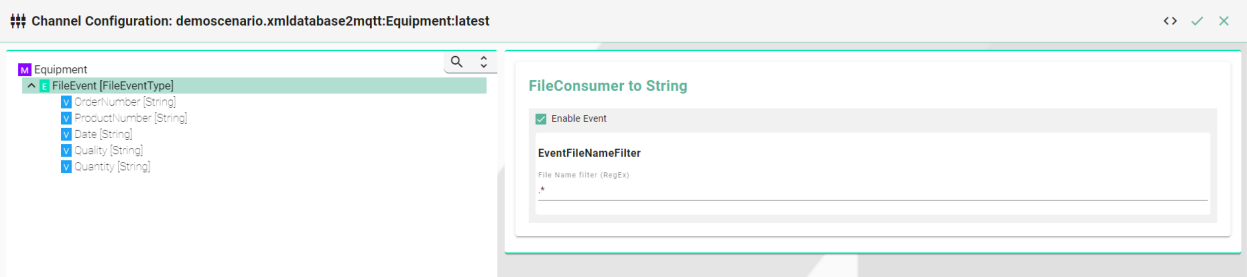
- Select the **Equipment** Information Model created previously.
- Select **File Reader (XML)** as Channel Type.

2. Configuration:

- Specify paths to following folder:
  - InFolder
  - ProcessFolder
  - OutFolder
  - ErrorFolder



- Select the Event to configure the *FileNameFilter*



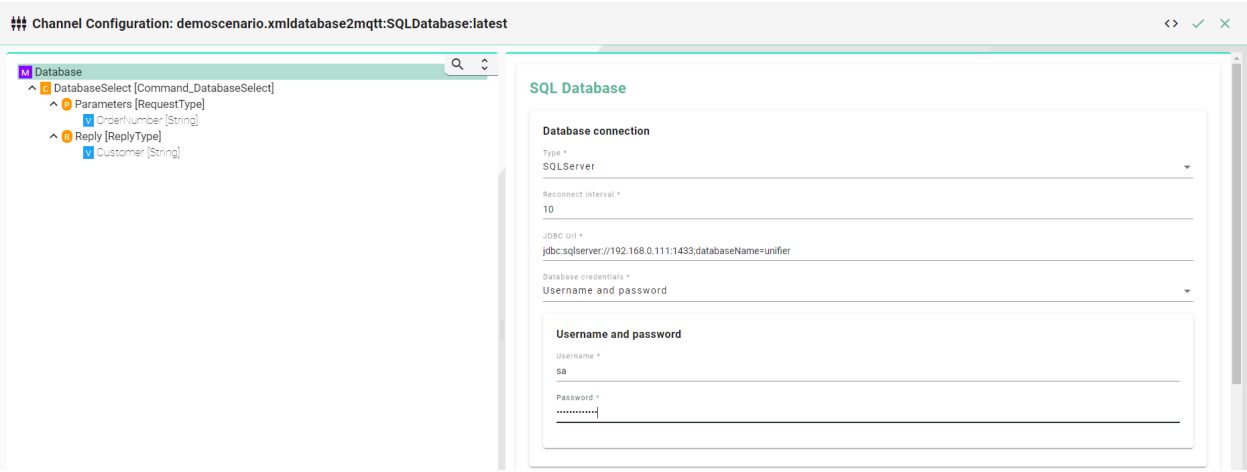
4.2 Communication Channel - SQL Database

1. Create a SQL Database Channel:

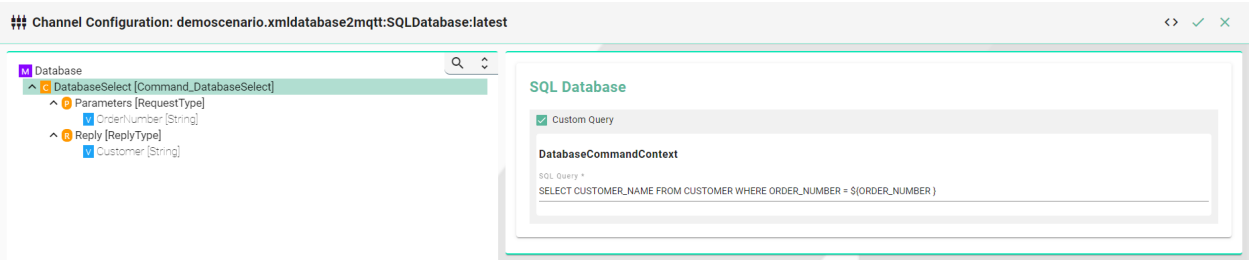
- Select the **Database** Information Model created previously.
- Select **SqlDatabase** as Channel Type.

2. Configuration:

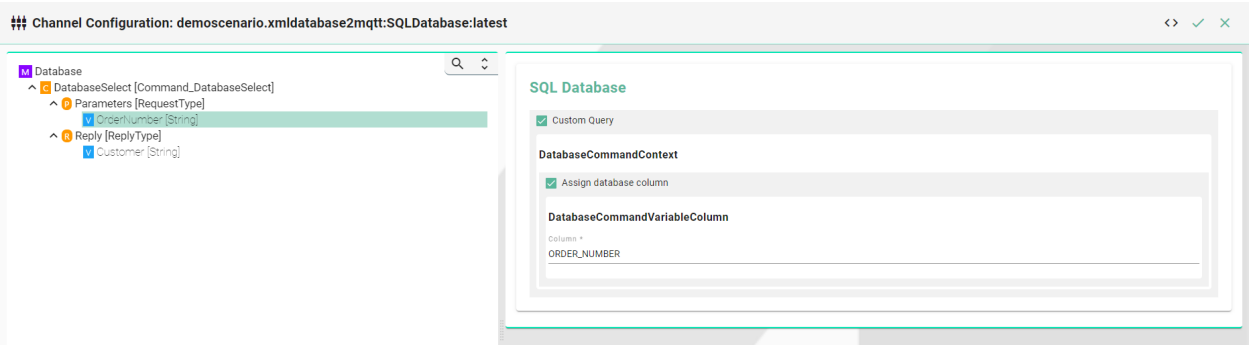
- Database Connection:
  - Select the database type **SQLServer**.
  - Set the **JDBC Url**, according to the selected database type - `jdbc:sqlserver://192.168.0.111:1433;databaseName=unifier`
  - Enter the **username** and the **password** or select it from the Credential Manager.



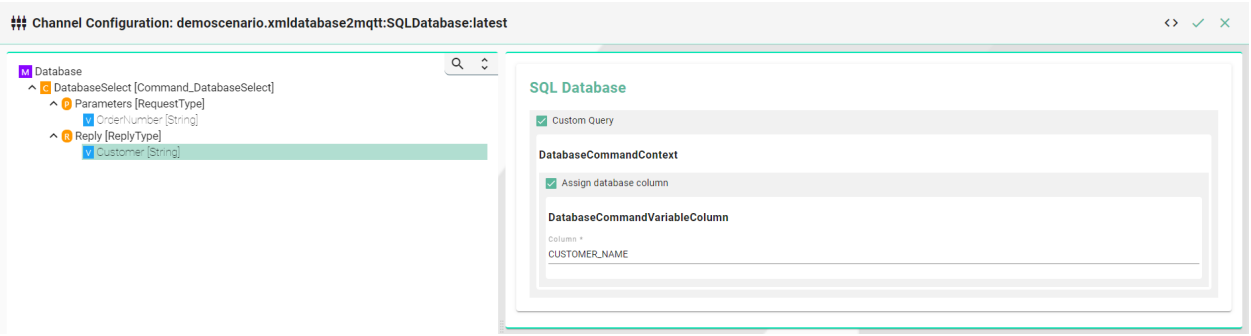
- Enter the SELECT query - `select CUSTOMER_NAME from DEMO_INTEGRATION_UC3_SCHEMA.CUSTOMER where ORDER_NUMBER = ${ORDER_NUMBER}`



- Enter the database column for the parameter **OrderNumber**.



- Enter the database column for the result variable **Customer**.



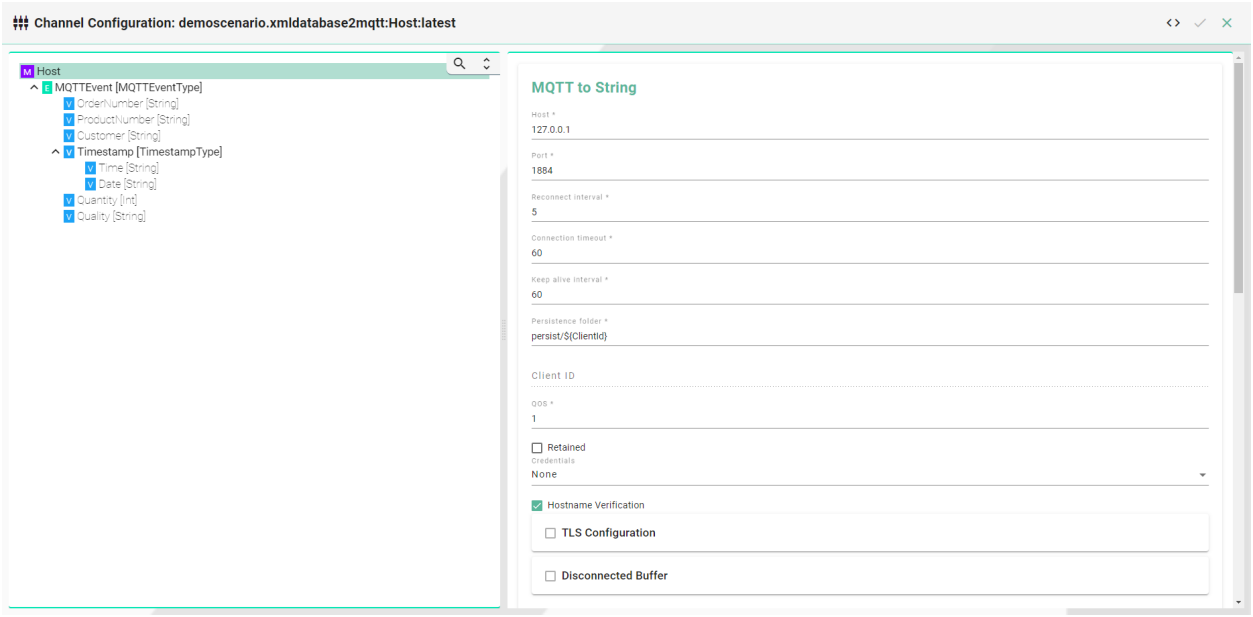
4.3 Communication Channel - MQTT

1. Create the MQTT Channel:

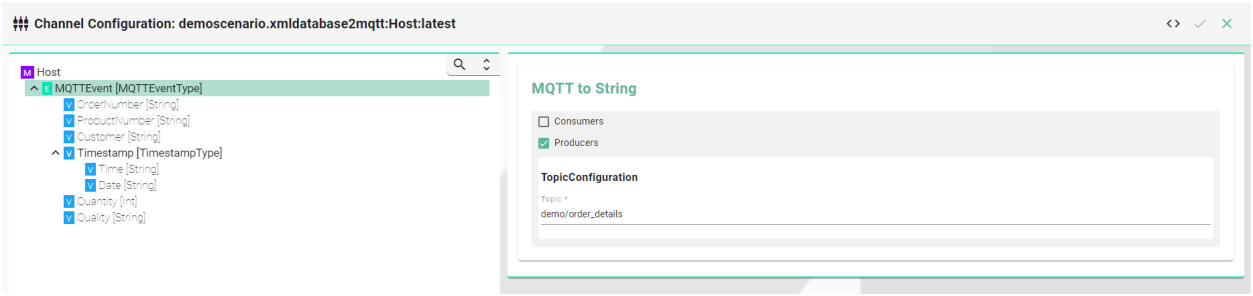
- Select the **Host** Information Model created previously.
- Select **MQTT (Json)** as Channel Type.

2. Configuration:

- Enter the **IP** of the MQTT Client.
- Enter the **Port** of the MQTT Client.



- Select the Event to enable the checkbox **Producers** and enter a **topic** name.



CHAPTER

FIVE

MAPPINGS

Create a Mapping with the Information Models created previously (Equipment, Database, and Host).

Create a Rule that executes a the SELECT query and sends the result with the other equipment data out via MQTT.

- Enter a **Rule Name**
- Select the **Edit Code** button.

Note: This scenario does not support the drag-and-drop functionality of the SMART**UNIFIER** Mapping due to type conversion and date formatting.

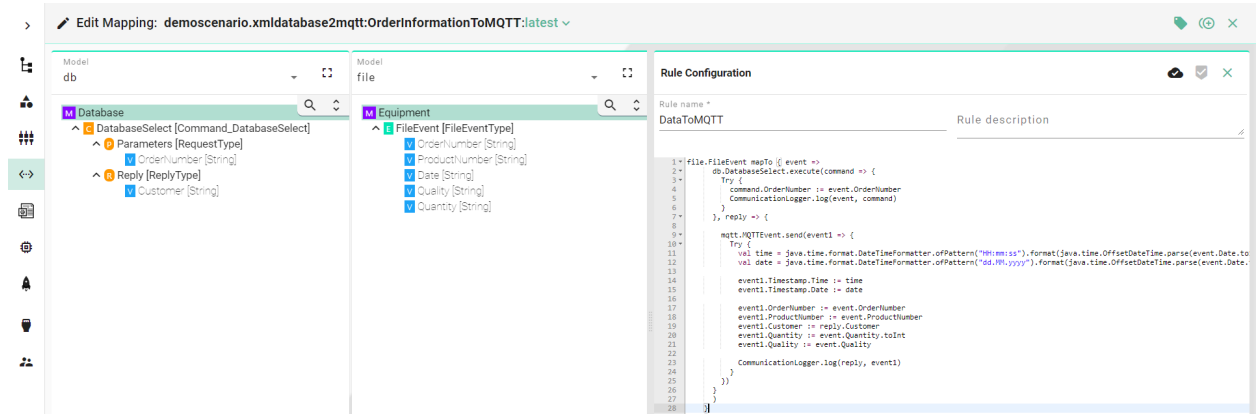
- Use the following code snippet and paste it into the Rule code editor:

```
equipment.FileEvent mapTo { event =>
  database.DatabaseSelect.execute(command => {
    Try {
      command.OrderNumber := event.orderNumber
      CommunicationLogger.log(event, command)
      }
  }, reply => {
    host.MQTTEvent.send(event1 => {

      Try {
        event1.Timestamp.time := java.time.format.DateTimeFormatter.ofPattern("HH:mm:ss").
↪format(java.time.OffsetDateTime.parse(event.date.value.toString))
        event1.Timestamp.date := java.time.format.DateTimeFormatter.ofPattern("dd.MM.yyyy").
↪format(java.time.OffsetDateTime.parse(event.date.value.toString))

        event1.OrderNumber := event.orderNumber
        event1.ProductNumber := event.productNumber
        event1.Customer := reply.Customer
        event1.Quantity := event.quantity.toInt
        event1.Quality := event.quality

        CommunicationLogger.log(reply, event1)
      }
```

(continues on next page)

(continued from previous page)

```
      }
      )
  }
  )
}
```



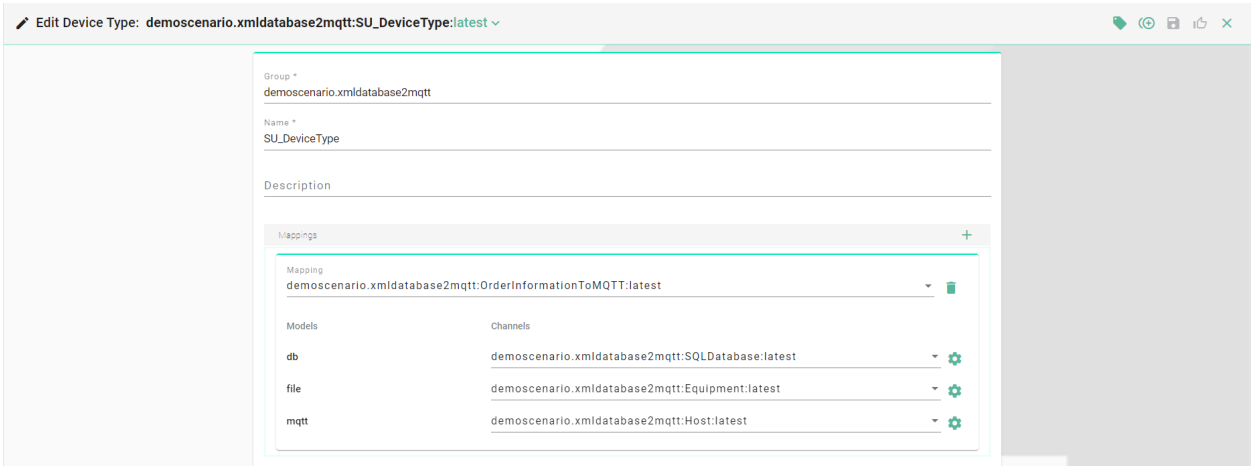
CHAPTER

SIX

DEVICE TYPE

Create a Device Type.

- Select the Mapping **EquipmentToHost** created previously
- Assign the Channels (Equipment, Database and Host (MQTT)) to their belonging Information Models.



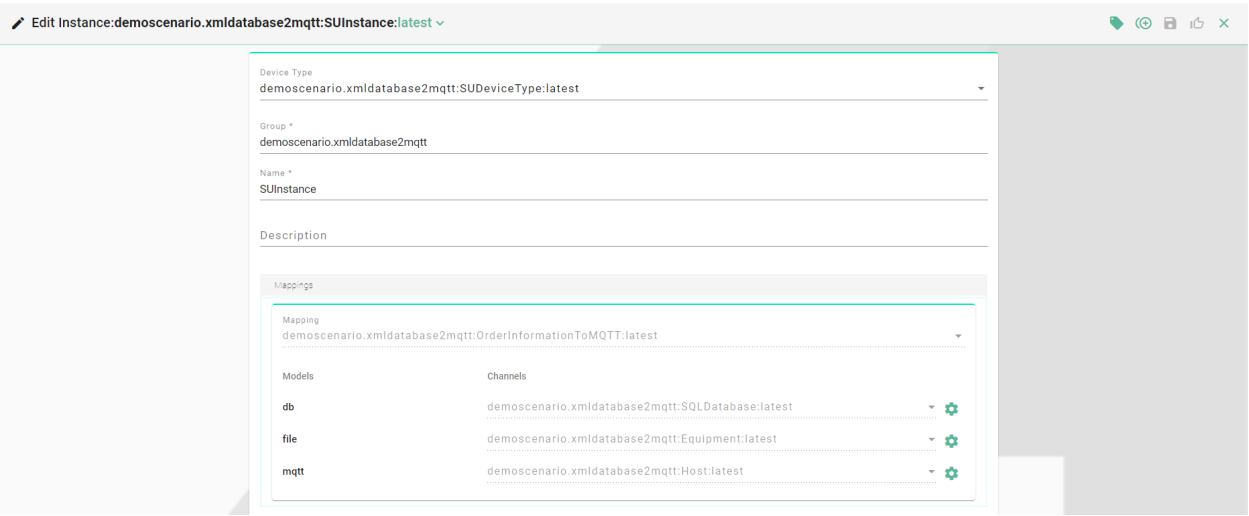
CHAPTER

SEVEN

INSTANCE

Create an Instance.

- Select the *Device Type* created previously.



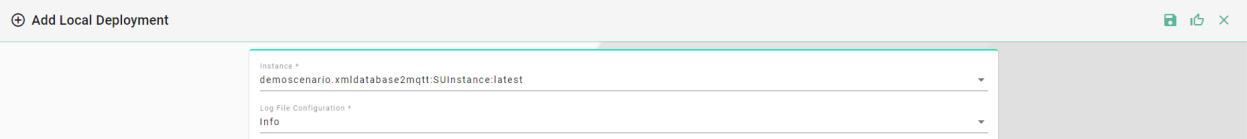
CHAPTER

EIGHT

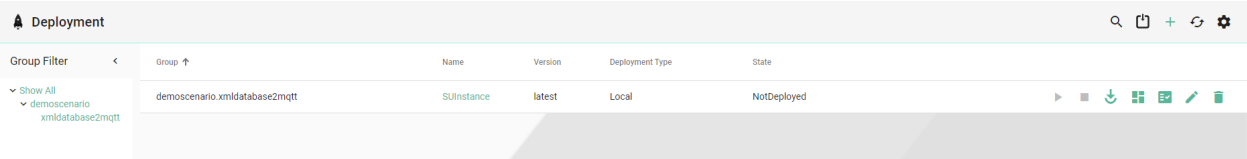
DEPLOYMENT

Create a new **Local** Deployment.

- Select the *Instance* created previously.
- Select the Log File Configuration *Info* (This defines the log detail).



Deploy and Start the Instance.



Execution

In order to send the data from the equipment with the customer information via MQTT, move the XML-file into the specified **InFolder**.