# Quantification Engine: A Java Library for Morbidity-Based Forecasting of Medicinal Products

Author: Alexey D. Kurasoff

Affiliation: Bureau THETA, member of Medicinal Products Digital Solution Consortium

Date: Jun 16, 2025

## Contents

# Abstract

Forecasting helps health facilities avoid running out of medicines or overstocking and wasting them. This paper introduces Quantification Engine—a free, open-source Java tool that uses disease incidence data to estimate the quantity and timing of medicine needs using disease incidence data.

# 1 Introduction

Running out of medicine or allowing it to expire is more than a planning error — it risks patient health and wastes valuable resources. Forecasts lose accuracy over time as conditions unexpectedly shift. Yet traditional tools, such as Excel workbooks, are not designed to adapt to dynamic, real-time data. The Quantification Engine supports on-demand recalculation of medicine forecasts based on the most current information, including updated treatment data and delayed procurement orders.

For large procurement organizations, maintaining a weekly or even daily forecast is essential to avoiding substantial financial losses and ensuring operational stability and trust.

# 2 Foundation

## 2.1 Method

There are two scientifically validated methods:

- Consumption-based forecasting uses historical usage data.
- Morbidity-based forecasting uses disease burden and population health trends.

The Quantification Engine relies on morbidity data, which offers a more stable and consistent basis for forecasting than historical statistics.

## 2.2 Forecasting period

The Forecasting Period is the specified time frame over which the expected demand for a medicinal product is estimated, based on current morbidity data, morbidity trends, known inventory, and incoming batches.

## 2.3 Morbidity data and morbidity trends

For the Quantification Engine, the morbidity data is the number of cases that begin the assigned treatment in a defined month. The current morbidity data are cases that started before the forecasting period and continue inside the forecasting period. The morbidity trends refer to cases that are expected to begin within the forecasting period.

## 2.4 Inventory and incoming batches

A batch is a specific quantity of a medicinal product produced during a single production cycle and identified by a unique batch code. All units in the batch have the same composition, quality, and expiration date. The Quantification Engine uses batch-level inventory tracking to comply with standard pharmaceutical practices.

Inventory refers to medicine batches that are physically available before the start of the forecasting period. Incoming batches arrive during the forecasting period. Each batch becomes usable in the days after its availability date.

## 2.5 Implementation

The Quantification Engine calculates daily forecasts for each medicine within the forecasting period:

**1. Starting stock**

The quantity of medicine available at the beginning of the day.

- On the first day, this equals the initial inventory (batches available before the start of the forecasting period).
- On subsequent days, it equals the previous day's End-of-Day Stock.

**2. Daily Needs**

The required quantity of medicine for the assigned treatments.

**3. Quantity Dispensed** is determined by:

- If Starting stock≥ Daily Needs, then:
  → Quantity Dispensed = Daily Needs
- Otherwise:
  → Quantity Dispensed = Starting Stock

**4. Missed Quantity**

The unmet portion of demand: Missed Quantity = Daily Needs – Quantity Dispensed

**5. End-of-Day Stock**

The stock remaining after dispensing: End-of-Day Stock = Starting Stock – Quantity Dispensed + quantity arrived

# 3 Input data

## 3.1 Format

The input data for forecasting conforms to the XML schema definitions specified in Forecast.xsd and DataTypes.xsd, located in the src/main/resources directory. Each XML element includes a description within annotation tags. The file bin/test_data/Fictitia_PLANNED.qtb provides an example of input data formatted for direct use.

## 3.2 Forecasting Period

The dates below define the specific period during which forecasting calculations are made.

| XML Element | Quantification Data |
|---|---|
| <element name="iniDate" type="date"> | First day of the forecasting period (inclusive) |
| <element name="endDate" type="date"> | End of the forecasting period (inclusive) |

## 3.3 Morbidity data and morbidity trends

Morbidity data showing the number of patients each month and the assigned treatment for each. An assignment consists of:

- Begin date.
- End date.
- The amount of medicine (dosage).
- How substantial the medicine is (strength).
- How often should it be taken (schedule)?

To use in the Quantification Engine, a morbidity data element must be encoded in XML.

| XML Element | Quantification Data |
|---|---|
| <complexType name="ForecastingRegimen"> | Morbidity Data Element |
| <element name="regimen" type="qgen:Regimen"> | The assigned treatment. |
| <element name="casesOnTreatment" type="frcst:MonthQuantity" maxOccurs="unbounded" minOccurs="0"> | The current morbidity data are cases that began before the forecasting period and continue inside the forecasting period. The start of treatment is presumed to be the first day of the month. |
| <element name="newCases" type="frcst:MonthQuantity" maxOccurs="unbounded" minOccurs="0"> | The morbidity trends refer to cases that are expected to begin treatment within the forecasting period. The start of treatment is presumed to be the first day of the month. |

## 3.4 Inventory and incoming batches

This information usually comes from Logistics Management Information Systems (LMIS), which are tools that help track medicine supplies.

A medicine definition consists of:

- International non-proprietary name – e.g., Acetylsalicylic Acid
- Abbreviation – e.g., ASA
- Strength for a unit – e.g., 500 mg
- Form – e.g., film-coated tablet
- List of batches in the inventory
- List of incoming batches – e.g., orders
- Results of quantification – not used for input data; see result definitions below
- Attrition coefficient for enrolled cases
- Attrition coefficient for expected cases

A batch definition consists of:

- Batch number
- A medicine definition
- The amount of medicine left
- The amount of medicine has expired
- Availability date
- Expiration Date

The attrition coefficient comes from local regulations and fine-tunes a medicine's needs forecast after the calculation:

- If the number is less than 100%, it shows that some treatments were stopped early or not used. This is called attrition, and the calculation result should be reduced.
- If the number is more than 100%, it means some extra medicine might be wasted. This is called wastage, and the calculation result should be increased.

To use in the Quantification Engine, a batch and a medicinal product must be encoded in XML as noted in the table below.

| XML Element | Quantification Data |
|---|---|
| <complexType name="ForecastingMedicine"> | A medicine definition |
| <element name="batchesToExpire" type="frcst:ForecastingBatch" maxOccurs="unbounded" minOccurs="0"> | List of batches in the inventory |
| <element name="orders" type="frcst:ForecastingOrder" maxOccurs="unbounded" minOccurs="0"> | List of incoming batches – e.g., orders |

## 3.5 Delivery parameters (advanced)

To recommend a medicine delivery order, three things are needed:

- Lead Time – how many months it takes from placing the order to getting the medicine
- Minimal stock – minimal reserve in months of dispensing
- Maximal Stock – maximal reserve in months of dispensing

These parameters control the date of a proposed delivery order and the quantity of medicine recommended for delivery. The delivery order proposal is an advanced feature of

the Quantification Engine. You can learn more about it later in this manual or check the source code *org.stoptb.quantbcalc.Calculator* class.

| XML Element | Quantification Data |
|---|---|
| <element name="minStock" type="int"> | Minimal stock that should be on hand. The parameter's value represents the number of months of future consumption. |
| <element name="maxStock" type="int"> | Maximal Stock that should be on hand. The parameter's value represents the number of months of future consumption. |
| <element name="leadTime" type="int"> | Months between placing an order and receiving the goods. |

## 3.6 Validation rules

Quantification Engine validates morbidity data, treatment regimens, medicine inventory, and incoming batches according to the following rules:

- Each treatment regimen must have at least one defined treatment case.
- All medicines used in treatment regimens must exist in the inventory.
- The inventory should not include medicines that are not part of any treatment regimen.
- All medicines in incoming batches must exist in the inventory.

# 4 Result

## 4.1 Format

The forecasting result data conforms to the XML schema definitions specified in Forecast.xsd and DataTypes.xsd, located in the src/main/resources directory. Each XML element includes a description within annotation tags.

The /bin/result.xml example file serves as a reference implementation. This file shows daily medicinal products and daily regimen forecasts.

## 4.2 Medicinal product forecast

For each day in the forecasting period, a medicinal product forecast shows how much is used for treatment, how much is missing (stockout), how much is expired, and how much is expected to arrive.

A daily forecast consists of:

- Month – the calendar month and year to which the result applies (e.g., March 2019)

- Start day – the date in the month (e.g., 5)
- End day – same as the start day
- Enrolled cases in treatment (e.g., 20)
- Expected cases in treatment (e.g., 40)
- Missed quantity – stockout (e.g., 0)
- End-of-day quantities in batches
- Batches will come on this day

| XML Element | Quantification Data |
|---|---|
| <complexType name="ForecastingMedicine"> | A medicinal product. Refer to the /bin/result.xml. |
| <element name="results" type="frcst:ForecastingResult" maxOccurs="unbounded" minOccurs="0"> | List of daily forecasts |

## 4.3 Treatment regimen forecast

A patient's treatment is a sequence of prescriptions. Each prescription may consist of multiple medication assignments for a defined period. These prescriptions are often referred to by clinicians as phases, such as intensive, continuous, and others. In many cases, treatment includes only an intensive phase. National treatment guidelines recommend the use of regimens based on scientific evidence and countrywide approved medications.

A daily treatment regimen forecast consists of:

- Month – the calendar month and year to which the result applies (e.g., March 2019)
- Start day – the date in the month (e.g., 5)
- End day – same as the start day
- Cases on treatment in intensive, continuous, and additional phases
- Medicines dispensed

| XML Element | Quantification Data |
|---|---|
| <complexType name="ForecastingRegimen"> | A treatment regimen. Refer to the /bin/result.xml. |
| <element name="results" type="frcst:ForecastingRegimenResult" maxOccurs="unbounded" minOccurs="0"> | A daily treatment regimen forecast |

# 5 Extra features

## 5.1 Aggregation for reporting

To aggregate quantification results from daily to monthly, the Quantification Engine provides the class *org.stoptb.quantbcalc.Calculator*. Details of the aggregation logic can be found in the *calculate* method, specifically in the section under the comment *"//Advanced Calculation."* You are welcome to include this aggregation step after the last forecasting step in the example provided.

## 5.2 Ordering scenarios

To implement different ordering scenarios, the Quantification Engine provides the class *org.stoptb.quantbcalc.Calculator*. Details can be found in the method *calculate*, specifically in the lines under the commentary "*// For each month, for each medicine, calculate monthly deliveries and estimate*." Feel free to include ordering scenarios after the *aggregation for the reporting* step in the example provided.

# 6 Implementation and Integration

The Quantification Engine has not been published in the Central Maven repository. To use it in your Maven or Gradle projects, upload the source code and integrate it into your environment.

# Conclusion

The Quantification Engine transforms medicinal product forecasting from reactive guesswork into a structured, proactive, and evidence-based planning process.

By integrating morbidity data, inventory levels, incoming batches, and attrition coefficients, it empowers supply chain teams to prevent stockouts, reduce waste, and strengthen public health outcomes.

# Keywords

Medicinal product forecasting, morbidity data, Java library, supply chain optimization, LMIS, HMIS, DHIS2, open-source health IT, stock management, regulatory benchmarks