Quantification Engine. Quick Start Guide

The Quantification Engine is the definitive open-source software library for empowering health information systems with precise, validated forecasts of medicinal product needs driven by robust, data-backed morbidity models. Built for pioneers in health digitization, the Quantification Engine delivers a powerful, fully customizable decision-support engine that adapts seamlessly to local practices and priorities.

Contents

Who Is This Guide For? 2

What You Need Before You Start? 2

Example: run qtbengine-4.5.0.jar 2

Understanding XML Data (For Curious Developers) 3

(Optional) Create Your Forecast Entry Point 3

Why Use This Example? 3

Final Thoughts 4

Who Is This Guide For?

Anyone who wants to empower a hospital, logistics, or analytical IT system with a proven medicinal product forecast. It will be beneficial for college students studying computer science, biomedical informatics, public health, or health information technology.

What You Need Before You Start?

Make sure you have the following installed:

- A Windows or Linux operating system
- Java 21 (or higher)
- An IDE like Eclipse, IntelliJ IDEA, or Visual Studio Code (recommended)

Next:

- Download the latest version of the Quantification Engine from GitHub: https://github.com/Bureau-THETA/qtbengine/releases/latest
- Unzip the file into a folder on your computer.

Example: run qtbengine-4.5.0.jar

The qtbengine-4.5.0.jar file includes an example class you can run right away. The main file is org.theta.Example. It demonstrates how to use the forecasting engine using test data in /bin/test_data.

Here's what you can do:

- Run run_test.cmd to start forecasting right away.
- View the forecast output in result.xml.
- Edit the input data in /test_data using any text editor.
- Explore the Java code in the org.theta.Example class, especially the main() method.
- See how it loads data, calculates the forecast, and writes the result.

You'll also find XML schemas (Forecast.xsd, DataTypes.xsd) that define the data formats and rules. These help you understand how to structure the input and output files properly.

Understanding XML Data (For Curious Developers)

The system uses XML files to store and exchange data. These are structured using two schema files found in qtbengine/src/main/resources:

- Forecast.xsd: Defines the forecast input/output
- DataTypes.xsd: Defines additional data structures

Look inside the annotation tags for helpful documentation. This example is a great way to learn open-source XML data modeling for medical IT systems.

(Optional) Create Your Forecast Entry Point

You can create your way to trigger forecasts using different classes depending on the project:

Project	Purpose	Entry Point	Features
QuanTB – desktop application	Calculating forecasting on local workstations. Collecting forecasting input data for exchange	Method "execute" in the "org.msh.quanta.services.calc.ForecastingCalculation"	The rich user interface for input data. Monthly forecast and early warnings. Delivery orders Graphs and tables for forecast data
Calculator, the web application	Provides forecasting-related APIs for other web applications.	Method "calculate" in the "org.stoptb.quantbcalc.Calculator"	Monthly forecast Delivery orders
This example	Provide an easy and comprehensive example for a quick start	Method "calculateDailyForecast" in the "org.theta.services.BasicEngine"	Daily forecast

The suggested entry points offer helpful hints for developing your entry point. If you're a student, this is an excellent opportunity to practice hands-on skills in healthcare software development.

Why Use This Example?

- No setup hassles – Unzip and run.
- Learn by doing – Change inputs and see how the forecasts change.

- Explore the source – Dive into the Java code to see how everything works.
- Check your XML – Validate your data using the provided schemas.
- Customize it – Build your integration for your own project or research task.

Final Thoughts

This example provides a streamlined way to explore qtbengine, ensuring accurate medicinal product forecasting and optimized supply chain management. Developers can use the example as-is or modify the entry point to suit their project's specific requirements.

The Quantification Engine helps you forecast medicinal needs using real-world health data and Java programming. This guide provides a solid foundation, whether you're an experienced developer or exploring career paths in health IT.

Use this example to build your skills in:

- Forecast modeling
- Open-source development
- Java-based health informatics

This library is a practical Java forecasting engine for healthcare IT experts and students who want to make a difference.