

Forecasting module

Documentation

v2.2

Document Released: April 28, 2014

frePPLe bvba  
Woluwestraat 17  
1930 Zaventem  
http://frepple.com

**Table of Contents**

[1. Introduction 3](#_Toc386477740)

[1. Functional overview 4](#_Toc386477741)

[a. Data loading 4](#_Toc386477742)

[b. History correction 4](#_Toc386477743)

[c. Baseline forecast generation 4](#_Toc386477744)

[d. Forecast review 5](#_Toc386477745)

[e. Forecast profiling 5](#_Toc386477746)

[f. Forecast consumption 5](#_Toc386477747)

[2. User guide 5](#_Toc386477748)

[3. Modelling 6](#_Toc386477749)

[a. Input table Forecast 6](#_Toc386477750)

[b. Input table ForecastDemand 6](#_Toc386477751)

[c. Parameters 6](#_Toc386477752)

# Introduction

This document describes the order quoting module of frePPLe.

This module support the calculation of forecasted demand, their management and review of these values by planners and sales people, and the pre-processing of the forecast in a format that can be used in frePPLe’s planning algoritm.

# Functional overview

A forecasting process typically consists of the following sub processes.

1. Data loading
2. History correction
3. Baseline forecast generation
4. Forecast review and editing
5. Forecast profiling
6. Forecast consumption

XXX grafiek met overview van processen

## Data loading

In a first step the input data are loaded into the frePPLe database. The key input consists of the following data elements:

* Items and their hierarchy.
* Customers and their hierarchy.
* Forecast, which defines a combination of a customer and an item where forecast calculations are performed.
* Historical demand history

Except for the last data element, the data are normally all loaded automatically through data interfaces with external systems. No manual intervention by the planner is then required for this step.

## History correction

The demand history may contain some exception, one-off demands which are often called “demand outliers”. In order to avoid that such demand influence the calculation of the statistical forecast in the next step too much, the planner should review the recent demand buckets for such exceptional demands and correct them.

Here are some typical situations where such corrections are required:

* + Exceptional demands, aka outliers
  + Product revisions, sku1 -> sku2 -> sku3

Note that the calculation of the baseline forecast has a built-in threshold correction for demand outliers. This feature takes care of some demand outliers which weren’t corrected by the planner, but can never achieve the same quality as review and analysis by the planner.

## Baseline forecast generation

In this step the system will apply statistical techniques on the demand history and extrapolate it into the future buckets. This generates automatically a **baseline forecast**.

These calculations are fully automated, and no planner intervention is required in this step.

The following time series forecasting techniques are implemented:

* **Moving average – flat forecast**  
  This methods uses the average of the last N buckets as the forecast for each future period.  
  FrePPLe will automatically use this technique if the time series doesn’t contain enough values to apply any of the other techniques that follow.
* **Single exponential – flat forecast**  
  This technique assigns exponentially decreasing weight on the previous demand buckets: the most recent time bucket gets weight 1-α, the bucket before, xxx, and so on.   
  The average average of this demand in these buckets is used as the forecast for each future period.  
  FrePPLe will automatically select the value of the parameter α to achieve the lowest deviation between the forecast and actual demand.
* **Double exponential – trending demand**
* **Winters – seasonal demand**
* **Croston – intermittent demand**

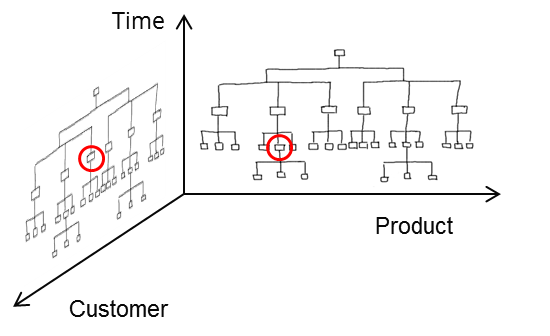
Each of these techniques is evaluated. The method which gives the lowest forecast error is automatically chosen to compute the baseline forecast. This evaluation is based on the symmetric (xxx ?) MAPE forecast error, where the forecast error in recent buckets is weighted more than the forecast error in older buckets.

## Forecast review

The forecast calculation in the previous step was fully automated. In this step users will review these numbers and apply their extra knowledge about the expected demand.

The review and correction of the forecast can happen at any level in the item and customer hierarchy. A sales manager might for instance be interested to review the forecast aggregated for all products by quarter in his region. A planner might want to review the forecast of each product in each month aggregated for all customers. The general manager will surely be interested in the total forecast across all items and all customers.

The forecast obtained at the end of this step is the **final forecast** that will be used for planning. The next 2 steps are automated calculations that bring the final forecast in a better structure for planning.



## Forecast profiling

Forecasting can happen in coarser time buckets than required for planning. For instance the sales people could forecast in monthly buckets. Such monthly buckets might not too inaccurate for planning the production. In such cases the forecast for the month can be profiled into weekly buckets, according to some predefined weights.

The forecast profiling is an automated process.

Xxx picture

## Forecast consumption

This step will subtract the orders already received from the total forecast. This is required to avoid double-counting the same demand.

The **net forecast** generated by this process is used as an extra demand stream by frePPLe’s planning algorithm.

The forecast consumption is an automated process.

* Net the customer orders from the gross forecast:
  + Gross forecast: 100
  + Orders already received: 20

> Net forecast: 80

# User guide

The forecast screen is the screen supporting all of the process steps described in the previous section.

Xxx screenshot

It shows the following data rows:

* **Orders total**This is a read-only row that is computed as the quantity available in the demand table.
* **Orders open**This is a read-only row that is computed from the data available in the demand table.
* **Orders planned**

This row show how much of the order book has been met.

* **Orders adjustment**

This row can be updated by the planner to correct demand outliers.

* **Forecast baseline**

This output row is the automatically computed forecast value by the system.

* **Forecast adjustment**

In this row the planner can enter adjustments to the baseline forecast.

* **Forecast total**

This is the sum of the forecast baseline and the adjustment.

* **Forecast consumed**

This row shows how much of the forecast has been consumed by the order book.

* **Forecast net**

This is the result of the forecast consumption. It represents the total forecast minus the demand that has already realized as customer orders.   
The sum of the rows “forecast consumed” and “forecast net” will always match the “total forecast” row.

* **Forecast planned**

In this row the planner can enter adjustments to the baseline forecast.

# Modelling

## Input table Forecast

## Input table ForecastDemand

## Parameters

The following extra parameters are introduced by this module.

|  |  |
| --- | --- |
| **Parameter** | **Description** |
|  |  |

My questions are:

* How to fill my data to the model environment? The old structure from version 0.9.2 is not working anymore.
  + I have monthly forecast quantities per item and sales regions.
  + I have also the effective sales (delivered, and open orders) per date, customer (contains to a sales region)  and item.

JDT> Forecast quantities go into the forecastdemand table. During planning the data will get merged into the forecastplan table as a forecast override, after which the forecastdemand table is emptied.

JDT> Sales orders go into the demand table. There is an extra field with the status: “open” or “closed”.

* + Are some fields calculated automatically by frePPLe?
    - How to start calculation in my installation?

JDT> If the forecast module is loaded in your file “init.xml”, the calculation happens automatically.

* + - Do I need to divide my qtys myself. Because solver or other functions are not working by loading model by ETL- process?
* How to use the forecast method in my case?
  + Write nettingQty to calendar-buckets and divide the qty to the numbers of days the bucket includes
  + What parameters are necessary for this ‘simple’ case?

JDT> The forecast method is used for the statistical forecast calculations.  You can leave it at the default value of “auto”