

---

# Memorandum

## eResept Forskrivningsmodul

### Thula – Nordic Source Solutions

Subject | The eResept PM Load Tester

Date | 10.4.2012

Author | Atli Sturluson

Distribution | **Helse Vest**

File name | Memo - eResept - 11-04-2012 - The eResept PM Load Tester.docx

---

Copyright © 2012 Thula

The document is only intended for Thula personnel and customers.  
Style and contents are confidential.

## 1 Background

The eResept PM Load Tester is a command-line program that can simulate activity from multiple PM clients (users) to put load on the PM server.

## 2 Using the simulator

The simulator starts by loading a random set of patients, prescriptions and other data from the database, which is then used for the simulation. Therefore, the database used must contain some data like patients and prescriptions. And of course, the simulator must **not** be run on a production database, because it will change the data, e.g. add new prescriptions and renew existing prescriptions.

### 2.1 The input file

The Load Tester takes as input an XML file that defines which types of operations and how many users should be simulated. It consists of a root element which contains a list of "Usecase" elements. Each "Usecase" element has three attributes: Type, Count and SecondsBetweenRuns, as explained below.

Sample input file:

```
<LoadTests RunTime="00:01:30">
  <Usecase Type="StartInbox" Count="10" SecondsBetweenRuns="10"/>
  <Usecase Type="LesVarslinger" Count="10" SecondsBetweenRuns="10"/>
  <Usecase Type="OpenPatientAndRenewPrescriptions" Count="10" SecondsBetweenRuns="10"/>
  <Usecase Type="OpenPatientAndCreatePrescription" Count="10" SecondsBetweenRuns="10"/>
</LoadTests>
```

The "RunTime" attribute defines for how long the test should run. More precisely put, no new runs are started after this time has elapsed. The simulation may run a little longer because it will continue until all instances have finished running.

Each "Usecase" has a "Type" attribute that defines which operation should be simulated. There are four types of use cases defined:

- **StartInbox:** Simulates a call to the EPJ API StartInbox.
- **LesVarslinger:** Simulates a call to the EPJ API method LesVarslinger.
- **OpenPatientAndRenewPrescriptions:** Simulates opening a patient (i.e. calling EPJ API method StartPasient), renewing a random number (1 to 5) of prescriptions, signing them and sending to RF.
- **OpenPatientAndCreatePrescription:** Simulates opening a patient (i.e. calling EPJ API method StartPasient), creating a new prescription, signing it and sending to RF. This includes creating and sending Helfo application for around 10% of the prescriptions (randomly selected).

The "Count" attribute defines how many "users" should be simulated and the "SecondsBetweenRuns" attribute defines how many seconds to wait after one invocation of the use case has completed until starting the next run.

### 2.2 Running the simulator

To run the simulator, open a command line window and run:

```
FM.LoadTester <epj username>/<epj password> <inputfile> <outputfile>
```

where <epj username> and <epj password> are the username and password of an existing EPJ user, <inputfile> is the input XML file and <outputfile> is a path to the file that the results should be written to. Example:

```
FM.LoadTester john/pass123 sampleinputfile.xml "c:\temp\load tester output.csv"
```

### 2.3 The output file

When the simulation finishes, the results are written to a semicolon-separated text file. Each line of the file has the following format:

<usecase type>;<timestamp>;<elapsed time>

<usecase type> is one of the four usecase types defined above, <timestamp> is the time when the result was recorded (i.e. when the run ended) and <elapsed time> is the time it took to run the usecase.

For example:

LesVarslinger;13:59:29.514;4,8821313

Here, the LesVarslinger usecase was run. It finished running at 13:59:29.514, after taking around 4,88 seconds to run. This list is sorted by the usecase type and the timestamp.

At the end of the file, there is a line containing only a hash sign “#” and then one line for each usecase type with summary information in the form

<usecase type>;<average>;<standard deviation>

i.e. the average and standard deviation of the elapsed time for each usecase type.

### 3 Sample simulation run

To test the performance of the PM server under heavy load, a series of simulation runs were performed, with increasing load and the results put in a graph for easy viewing.

The test was run on a “decent” machine (Intel i7 processor, 8 GB memory, SSD drive) with the test client, PM server and the database server all running on this same computer. The database used for the tests contained around 16000 patients and 25000 prescriptions.

The graph below shows how the average execution time of each usecase type increases when more load is put on the server. The X axis is the number of concurrent "users" running each type of usecase. For this test, the following input file was used (where Count="X" corresponds to the x axis value):

```
<LoadTests RunTime="00:01:30">
  <Usecase Type="StartInbox" Count="X" SecondsBetweenRuns="10"/>
  <Usecase Type="LesVarslinger" Count="X" SecondsBetweenRuns="10"/>
  <Usecase Type="OpenPatientAndRenewPrescriptions" Count="X" SecondsBetweenRuns="10"/>
  <Usecase Type="OpenPatientAndCreatePrescription" Count="X" SecondsBetweenRuns="10"/>
</LoadTests>
```

From these results, it seems that the performance is not "falling off a cliff" for high load, but showing a steady decline as more load is put on the server.

