

Project Title: **Μετάφραση και διάδοση του ηλεκτρονικού ιατρικού φακέλου ανοικτού κώδικα OpenMRS, για την βελτίωση παροχής των υπηρεσιών Υγείας**

Project Acronym: **OpenMRS translationd**

Παραδοτέο: Εγκατάσταση του συστήματος OpenMRS

Deliverable: Setup of the OpenMRS system

Contents

Executive Summary	2
OpenMRS introduction.....	3
Installation and initial setup	4
Step 1 - Install Firefox.....	4
Windows.....	4
Step 2 - Install Java.....	4
Windows.....	4
Ubuntu.....	5
Other Operating Systems	5
Step 3 - Install Tomcat	5
Windows.....	5
Other operating systems.....	6
As a Debian package.....	7
Jetty as an alternative to Tomcat	8
Security Enhancements	8
Step 4 - Install MySQL.....	9
Other Operating Systems	9
Step 5 - Deploy OpenMRS	9
Step 6 - Configuration.....	10
Step 7 - Start Using OpenMRS.....	10
Translation of the OpenMRS.....	10

Executive Summary

In this deliverable we are going to describe the steps regarding the installation and translation of the OpenMRS system

OpenMRS introduction

OpenMRS¹ is an electronic medical record system (EMR), designed for use in the developing world and first established in 2004. Today, the system has evolved into a medical informatics platform used on every continent, supporting health care delivery and research in an extremely wide variety of contexts.

Our world continues to be ravaged by pandemics of epic proportions, as untold millions of people are infected with diseases such as HIV/AIDS, multi-drug resistant tuberculosis, malaria, and many others. Many of these infections occur in developing countries, where lack of education and resources contribute to scores of preventable deaths. Prevention and treatment interventions on this scale require efficient information management, which is particularly critical as clinical care must increasingly be entrusted to less skilled providers. Whether for lack of time, lack of money, or no access to software developers, most health care programs in developing countries manage their information with simple spreadsheets or small, poorly designed databases—if they have any electronic infrastructure at all. Most health care records in the developing world are still maintained on paper.

As a response to these challenges in developing countries, OpenMRS was created as a medical record platform—a rising tide which we hope will lift all ships. It is designed to offer a better tool for information management, but also to reduce unnecessary, duplicate efforts. In the years since its inception, the OpenMRS community has grown from a handful of organizations to a massive collaborative effort by both groups and individuals, all focused on creating medical record systems and a corresponding implementation network that allows self-reliance in system development, even in resource-constrained environments.

Since its beginning, OpenMRS has been based on the principles of openness and of sharing ideas, software and strategies for deployment and use. The system is designed to be usable in very resource-poor environments and can be modified with the addition of new data items, forms and reports without the need to write complicated application code. It is intended as a platform that organizations can adopt and modify, avoiding the need to develop a system from scratch.

And indeed, organizations around the world are doing just that. OpenMRS is now in use in clinics in Argentina, Botswana, Cambodia, Congo, Ethiopia, Gabon, Ghana, Haiti, Honduras, India, Indonesia, Kenya, Lesotho, Malawi, Malaysia, Mali, Mozambique, Nepal, Nicaragua, Nigeria, Pakistan, Peru, Philippines, Rwanda, Senegal, South Africa, Sri Lanka, Tanzania, The Gambia, Uganda, United States, Zanzibar, Zimbabwe, and many other places. This work is supported by many individuals and organizations, including international and government aid groups, NGOs, and for-profit and non-profit corporations.

OpenMRS is not only in use in many different places, but it is also being used to meet many different needs. In Kenya, it is used to support health care delivery for hundreds of thousands of patients at a network of over 50 clinics—some connected by typical networks, but many where the connection requires offline synchronization to external storage that can be physically transported between sites! Another NGO uses a central OpenMRS server connected to clinics in multiple countries via satellite Internet connections. In Malawi, creative individuals with a talent

¹ www.openmrs.org

for technology have created a mobile cart running OpenMRS that physicians roll around their clinic, interacting with the system using a touchscreen. In Rwanda, the national ministry of health has worked to roll out a connected national health care system using OpenMRS. In the United States, OpenMRS is used to track patients at large sporting events, for mobile providers of health care to homeless people, and as a personal health record that allows cancer patients to share treatment and home health care information with caregivers and family members.

In the last several years, use of mobile technology has increased dramatically, particularly in the developing world. In some developing countries, there are more mobile phones than people! Facilitated by other open source projects, OpenMRS can be integrated with SMS messaging, allowing community health workers to add information about adherence to medication regimens to a patient's record, as they make rounds through villages in rural Africa. Elsewhere, mobile phone applications are used to guide these community volunteers in home-based HIV testing and counseling, enrolling prospective patients from the comfort of their own homes.

Besides clinical care, the platform can also be used in research settings. In the United States, OpenMRS has been used both in training medical informatics students, as well as in conducting various research projects in the fields of public health. In Peru OpenMRS is used as the research database for a large study of drug resistant tuberculosis funded by the US National Institutes of Health. Because the system has been designed as an extensible platform, it is very easy for researchers to adapt OpenMRS to do what they need.

Installation and initial setup

There are some steps² that we should follow in order to install and setup the OpenMRS system. The steps are the following:

Step 1 - Install Firefox

Windows

1. Download the latest stable release of Firefox and run installation program
2. Accept the license agreement
3. Select standard mode for installation or Custom, make sure the application is installed on the default directory. C:\Program Files (x86)\Mozilla Firefox

Step 2 - Install Java

Windows

1. Download the latest stable release of the [Java Runtime Environment \(JRE\)](https://www.oracle.com/technetwork/java/javase-downloads-1344942.html)

² www.openmrs.org

- For the standalone version you will only need the Java Runtime Environment (JRE) not Java Development Kit (JDK)
- Minimum version required is Java 6 although it is recommended to install Java 7
- Java 8 is not currently supported To download the application use the this link <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Accept the license agreement and make sure you download the correct file for your windows version, whether its x32 or x64 bit.

Note: If you are unsure about the version please follow this guide from Microsoft <http://windows.microsoft.com/en-US/windows7/find-out-32-or-64-bit>

- It is recommended to download the executable file (.exe) for a simpler run
- Execute the downloaded file, accept the license agreement and follow the instructions in the wizard, installing in default installation directories.

Ubuntu

You can install the OpenJDK on it's own as a package
`sudo apt-get install openjdk-6-jdk`

or automatically as a dependency of Tomcat
`sudo apt-get install tomcat6`

Other Operating Systems

1. Download the latest stable release of the [Java Runtime Environment \(JRE\)](#)
2. Run the installer (or unzip the contents, whichever is needed)
3. Accept the license agreement

Step 3 - Install Tomcat

- [Java](#) must be installed before installing Apache Tomcat.
- There are issues with versions of Tomcat later than 6.0.29 that have yet to be resolved. Installation through a package manager is not recommended as this will likely install a later version
- With OpenMRS 1.8 it is necessary to increase the Tomcat Permgen memory after installing Tomcat but before deploying OpenMRS. More information: <https://wiki.openmrs.org/display/docs/Troubleshooting+Memory+Errors>

Windows

1. Download [Tomcat 6.0.29](#). You can use the [exe](#) version, which installs Tomcat as a service or the [zip](#) archive.
- a. Execute the file and install running the default settings o Accept the license agreement

1. Accept default destination folder
2. Accept HTTP/1.1 Connector Port 8080
3. Set Administrator login (username/password)
4. Accept the Java directory detected
5. Select Install Tomcat# After installation is complete you will need to change users roles by following this directory on your windows explorer
 - a. C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf
 - b. Locate the file "tomcat-users.xml" and try to open it.
- i. Most likely your operating system will fail to detect the application that opens the file so make a right-click on the file then select down the menu Open With > Notepad
- ii. You will notice that a text editor will show up then locate this character set <tomcat-users> The character set is located on line 18 of the file.
6. Open the Tomcat users file (e.g. C:\Program Files\Apache Software Foundation\Tomcat 6.0\conf\tomcat-users.xml) in a text editor.
7. Create a new user called *admin* with the roles *admin*, *manager* and *manager-gui*. This file should be protected so you will need to open it as Administrator (right-click on your text editor and select "Run as administrator")

```
<user name="admin" password="XXXXXX" roles="tomcat,admin,manager,manager-gui"/>
```

Then save the file

1. Your operating system might bring an error message that indicates that you do not have sufficient privileges to save the file. Then it will ask you to save it in a different directory.
 - a. You need to save the file in the current directory, right-click on the file "tomcat-users" and click on Properties, at the bottom of the menu.
 - b. Navigate to the "Security" tab
 - c. Select the username you are currently using on the machine
 - d. Click the "Edit" button
 - e. Permissions table will allow you to edit your privileges as a user.
 - f. Click on Full Control then click OK and then OK again
 - g. Now, you should be able to edit and save the file in the same directory.

(Optional) If you've installed Tomcat as a service, you can configure it to start automatically when the computer boots:

1. Start > Settings > Control Panel > Administrative Tools > Services
2. Right Click "Apache Tomcat" > Properties > Set "Startup Type" to Automatic
3. Click Start or restart your pc

Other operating systems

1. Download the [zip](#) archive of [Tomcat 6.0.29](#)
2. Unpack the zip file to a suitable location such as /opt on Linux or /Library on Mac OSX

```
sudo useradd tomcat6

cd /opt

sudo tar zxvf apache-tomcat-6.0.29.tar.gz

sudo ln -s apache-tomcat-6.0.29 tomcat6

sudo chown tomcat6.tomcat6 apache-tomcat-6.0.29
```

Open the Tomcat users file (e.g. `/opt/tomcat/conf/tomcat-users.xml`) in a text editor. Create a new user called *admin* with the roles *admin*, *manager* and *manager-gui*. This file should be protected so you will need to open it as root (e.g. `sudo nano /opt/tomcat/conf/tomcat-users.xml`)

```
<user name="admin" password="XXXXXX" roles="tomcat,admin,manager,manager-gui"/>
```

As a Debian package

This is not recommended as it may install a version of Tomcat which is not compatible with OpenMRS.

1. Run the following command from a terminal

```
sudo apt-get install tomcat6
```

Open the Tomcat users file (e.g. `/etc/tomcat/tomcat-users.xml`) in a text editor. Create a new user called *admin* with the roles *admin*, *manager* and *manager-gui*. This file should be protected so you will need to open it as root (e.g. `sudo nano __/etc/tomcat/tomcat-users.xml`)

```
<user name="admin" password="XXXXXX" roles="tomcat,admin,manager,manager-gui"/>
```

Turn off tomcat security flag in `/etc/init.d/tomcat6` file: Find `"TOMCAT6_SECURITY=yes"` and change it to `"TOMCAT6_SECURITY=no"`

Create OpenMRS application data directory and make it writable by Tomcat: (so that the [runtime properties](#) file can be written by the webapp during initial startup)

```
sudo mkdir /usr/share/tomcat6/.OpenMRS
```

```
sudo chown -R tomcat6:root /usr/share/tomcat6/.OpenMRS
```

To start/stop/restart tomcat6, please type the following commands:

```
sudo service tomcat6 start
```

```
sudo service tomcat6 stop
```

```
sudo service tomcat6 restart
```

Jetty as an alternative to Tomcat

This is meant to run in a Linux environment.

1. Download the Jetty 7.4.5 tar.gz from [here](#). Don't download 7.5.4; it may not recognize the jdk that you have installed.
2. Unpack the tar file to your preferred directory (I usually use /usr/share/jetty)

```
sudo mkdir /usr/share/jetty
```

```
cd /usr/share/jetty
```

```
sudo mv /pathtojetty/jetty-distribution-(version).tar.gz .
```

```
sudo tar xzf jetty-distribution-(version).tar.gz
```

```
sudo mv jetty-distribution-(version)/* .
```

```
sudo rm -rf jetty-distribution-(version)
```

Now to make it start when you start the system and make Jetty a service

```
sudo cp bin/jetty.sh /etc/init.d/jetty
```

Edit /etc/init.d/jetty to include the following two lines after the comments so Jetty knows where your Java and Jetty directories are.

```
JAVA_HOME=(path to java)
```

```
JETTY_HOME=/usr/share/jetty //or where your jetty installation directory
```

Jetty is now officially installed and can be run as a service. Now you can run Jetty by using the following command. First put the openmrs.war in to /usr/share/jetty/webapps/ so Jetty will know to run the war.

```
sudo /etc/init.d/jetty start
```

Security Enhancements

- In newest versions of Tomcat(> version 7), by default HttpOnly flag will be set by the server. But in older versions of Tomcat, it needs to set this flag through a configuration. The HttpOnly flag is an additional flag that is used to prevent an XSS (Cross-Site Scripting) exploit from taking access to the session cookie. Because one of the most known ways of subjecting to an XSS attack is access to the session cookie, and to subsequently hijack the victim's session, the HttpOnly flag is a useful prevention mechanism where a client side script won't be able to access the session cookie from. To add the HttpOnly flag to session cookies in older versions of Tomcat, you need to edit the <TOMCAT_HOME>/conf/context.xml to add useHttpOnly="true" attribute as below:
<Context useHttpOnly="true">

```
<Manager pathname="" />
```



```
<Valve className="org.apache.catalina.valves.CometConnectionManagerValve" />

</Context>
```

- <https://issues.openmrs.org/browse/TRUNK-3941>

Step 4 - Install MySQL

Windows

- Download the latest MySQL installer using this [link](#)
- Run the install program (.msi)
- Accept the license agreement
- When given the option to update installer please do so
- Under Feature Selection select Full Installation Setup and select the right Architecture for your computer (32-bit / 64-bit)
- Click next and you will be shown a list of applications that you need in order to meet the requirements for installing all services. Make sure you satisfy all the requirements, if not, please install the missing applications on your machine.
- On the next configuration options select “Developer Machine”
- Leave all other settings to default
- Enter a username and password. Note: These will be the credentials for the user with root privileges. Do Not Forget the Password
- Click next and finish the installation.

Note: MySQL might fail to run as a service, for this you can manually start it by navigating to Start > Settings > Control Panel > Administrative Tools > Services. Then find the service called “MySQL”, right click > Properties then you can either click the “start” button or set “Startup Type” to automatic.

Other Operating Systems

1. Install the MySQL server package: **sudo apt-get install mysql-server**
2. Enter a root password

Step 5 - Deploy OpenMRS

- With OpenMRS 1.8 it is necessary to increase the Tomcat Permgen memory before deploying OpenMRS. More information: <https://wiki.openmrs.org/display/docs/Troubleshooting+Memory+Errors>
1. (In Windows) Ensure that Tomcat is started by checking to see if icon in the tray is green
 2. [Download the latest stable release of OpenMRS](#)

3. Navigate to <http://localhost:8080/manager/html> and enter your Tomcat administrator credentials (username and password chosen when installing Tomcat)
4. In the Tomcat Web Application Manager, enter the location of the downloaded [openmrs.war file](#) to deploy
 - a. The deployment could take some time while the file is copied to the folder C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps and decompressed
 - b. Note that the **OpenMRS.war** file is most easily downloaded with [Mozilla Firefox](#). Internet Explorer tries to open the file as a Zip file.
5. At the end of this process, the web page will refresh and **/openmrs** should be displayed under Applications. Apache Tomcat should also start the application (Running = True; and in Commands, Stop is underlined)

Note: Another way to do this is just unzipping the .war file directly under webapps folder in Tomcat and then restarting it. You will be able to access <http://localhost:8080/openmrs> and the installation wizard will appear.

Step 6 - Configuration

Unable to render {include}

 The included page could not be found.

Step 7 - Start Using OpenMRS

1. After you have finished configuring OpenMRS, **RELOAD** the application in Tomcat Manager.
2. Open <http://localhost:8080/openmrs>. You will see a login page. If you're using the OpenMRS standalone package, the page is at <http://localhost:8080/openmrs-standalone>.
- a. You will need to log in initially using the username and password you specified in [Step 6 - Configuring OpenMRS](#), substep 4. If you did not specify a username and password, try the default username **admin** and password **test** (both are in lowercase).
- b. Alternatively, while Tomcat is running you can start OpenMRS by entering <http://localhost:8080/openmrs/login.htm> (assuming 8080 is your port number for Tomcat; insert the appropriate port number if it is not 8080). For OpenMRS standalone package, you can start OpenMRS by entering <http://localhost:8080/openmrs-standalone/login.htm>.

Translation of the OpenMRS

In order to proceed to the translation of the openMRS system there are two steps that we are going to follow.

1. Adding or updating a translated language

In our case, cause there is no translation before for the Greek language, we are going to add a new translated language.

This is relatively easy. After downloading or saving **messages.properties**, rename a copy of it to the Greek language:

messages_xx.properties

where the "xx" is replaced with the two-letter, lowercase code from [ISO 639-1](#). Edit the file and modify all the phrases or words on the **right side** of the equals (=) sign to the new language. The left side of each line (left of the equals sign) must remain the same – they are needed by the system. You need to save the file as UTF-8 with BOM and leave the first line empty.

Put this new file into the **WEB-INF** folder of your OpenMRS installation and edit the **locale.allowed.list** global property in the OpenMRS administration section to list this new language code. You can now switch to that locale at the bottom of any OpenMRS page. If you cannot, then you need to restart OpenMRS.

2. Submitting the files to OpenMRS
[Create a new TRUNK issue in JIRA](#), our issue tracking system, and attach your modified **messages.properties** or new **messages_xx.properties** file. Explain what changes or the new language that you've added. Someone will review your updates or additions for inclusion in an upcoming OpenMRS release.